

# Strukturelle Analyse Web-basierter Dokumente



Vom Fachbereich Informatik  
der Technischen Universität Darmstadt  
genehmigte

## **Dissertation**

zur Erlangung des akademischen Grades Dr. rer. nat.

vorgelegt von Dipl.-Mathematiker Matthias Dehmer  
geboren in Wiesbaden

Tag der Einreichung: 25. Juli 2005

Tag der Disputation: 6. September 2005

Referenten:

Prof. Dr. Max Mühlhäuser, Darmstadt

Prof. Dr. Alexander Mehler, Bielefeld

Darmstädter Dissertationen D17



# Vorwort

Die vorliegende Arbeit entstand im Rahmen meiner Tätigkeit als Doktorand im Fachgebiet Telekooperation des Fachbereichs Informatik an der Technischen Universität Darmstadt.

Meinem Doktorvater Prof. Dr. MAX MÜHLHÄUSER danke ich für die große Freiheit, mit der ich fachlich das Thema bearbeiten und die Arbeit erstellen konnte. Dadurch, dass er mir alle Möglichkeiten innerhalb seines Fachgebiets zur Verfügung stellte und mich förderte, schaffte er die Voraussetzung für eine reibungslose Durchführung der Arbeit. Diese Unterstützung hat mir sehr geholfen. Auch menschlich verdanke ich ihm sehr viel, so dass ohne ihn die Arbeit in der von mir angestrebten Zeit nicht zustande gekommen wäre.

Prof. Dr. ALEXANDER MEHLER, der die Zweitgutachtertätigkeit übernahm, danke ich einerseits für die besonders gute und fruchtbare Zusammenarbeit während meiner Dissertationsphase. Unsere Zusammenarbeit im Rahmen von Publikationen und Diskussionen wirkte sich sehr positiv auf die Erstellung der Arbeit aus, so dass er maßgeblich die Qualität dieser Arbeit verbesserte. Weiterhin danke ich in diesem Zusammenhang Dipl.-Inform. RÜDIGER GLEIM, der im Rahmen dieser Arbeit mit großem Elan seine Diplomarbeit anfertigte. Damit unterstützte er mich stark mit Implementierungsarbeiten und anregenden Diskussionen.

Dr. FRANK EMMERT-STREIB danke ich zum einen für die äußerst gute und erfrischende Zusammenarbeit und zum anderen für wertvolle und konstruktive Hinweise, betreffend Kapitel (6). Dr. JÜRGEN KILIAN gebührt mein Dank für die Mithilfe zur Klärung grundlegender Konstruktionsmerkmale des Graphähnlichkeitsmodells, insbesondere bezüglich praktischer Aspekte der dynamischen Programmierung. Somit hat er wesentlichen Anteil am Gelingen des Kapitels (5), welches eine wichtige Grundlage für die Arbeit bildet. Dr. habil. ULRIKE BRANDT danke ich für die Diskussionen in der Anfangsphase meiner Arbeit.

Ganz besonders möchte ich meinem Vater WERNER DEHMER danken, der mich in der Endphase der Arbeit finanziell unterstützte. Insbesondere danke ich meiner Freundin JANA. Sie hat während der Erstellung der Arbeit viel Geduld und Verständnis aufgebracht. Für das sprachliche Korrekturlesen dieser Arbeit bedanke ich mich bei MARION DEHMER-SEHN M.A., Dr. SANDRA BOHLINGER, JULIA HINSKE, STEVE HINSKE, MONIKA LEHR-WLEKLINSKI, Dipl.-Inform. (FH) NICOLAS KALKHOF und Dipl.-Ing. JANA MÜNZNER. Dipl.-Inform. (FH) KARIN TILLACK danke ich für ihre Hilfe bei der Erstellung einiger Graphiken.

Darmstadt, im Juli 2005

MATTHIAS DEHMER



# Zusammenfassung

Im Zuge der web-basierten Kommunikation und in Anbetracht der gigantischen Datenmengen, die im World Wide Web (kurz: Web) verfügbar sind, erlangt das so genannte Web Mining eine immer stärkere Bedeutung. Ziel des Web Mining ist die Informationsgewinnung und Analyse web-basierter Daten auf der Grundlage von Data Mining-Methoden. Die eigentliche Problemstellung des Data Mining ist die Entdeckung von Mustern und Strukturen in großen Datenbeständen. Web Mining ist also eine Variante des Data Mining; es kann grob in drei Bereiche unterteilt werden: Web Structure Mining, Web Content Mining und Web Usage Mining.

Die zentrale Problemstellung des Web Structure Mining, die in dieser Arbeit besonders im Vordergrund steht, ist die Erforschung und Untersuchung struktureller Eigenschaften web-basierter Dokumente. Das Web wird in dieser Arbeit wie üblich als Hypertext aufgefasst. In der Anfangsphase der Hypertextforschung wurden graphbasierte Indizes zur Messung struktureller Ausprägungen und Strukturvergleichen von Hypertexten verwendet. Diese sind jedoch im Hinblick auf die ähnlichkeitsbasierte Gruppierung graphbasierter Hypertextstrukturen unzureichend. Daher konzentriert sich die vorliegende Arbeit auf die Entwicklung neuer graphentheoretischer und ähnlichkeitsbasierter Analysemethoden.

Ähnlichkeitsbasierte Analysemethoden, die auf graphentheoretischen Modellen beruhen, können nur dann sinnvoll im Hypertextumfeld eingesetzt werden, wenn sie aussagekräftige und effiziente *strukturelle* Vergleiche graphbasierter Hypertexte ermöglichen. Aus diesem Grund wird in dieser Arbeit ein parametrisches Graphähnlichkeitsmodell entwickelt, welches viele Anwendungen im Web Structure Mining besitzt. Dabei stellt die Konstruktion eines Verfahrens zur Bestimmung der strukturellen Ähnlichkeit von Graphen eine zentrale Herausforderung dar. Klassische Verfahren zur Bestimmung der Graphähnlichkeit beruhen in den meisten Fällen auf Isomorphie- und Untergraphisomorphiebeziehungen. Dagegen wird in dieser Arbeit ein Verfahren zur Bestimmung der strukturellen Ähnlichkeit hierarchisierter und gerichteter Graphen entwickelt, welches nicht auf Isomorphiebeziehungen aufbaut.

Oft wird im Rahmen von Analysen web-basierter Dokumentstrukturen das bekannte Vektorraummodell zu Grunde gelegt. Auf der Basis eines graphbasierten Repräsentationsmodells wird dagegen in dieser Arbeit die These vertreten und belegt, dass die graphbasierte Repräsentation einen sinnvollen Ausgangspunkt für die Modellierung web-basierter Dokumente darstellt. In einem experimentellen Teil werden die entwickelten Graphähnlichkeitsmaße erfolgreich evaluiert und die aus der Evaluierung resultierenden Anwendungen vorgestellt.



# Abstract

In the course of web-based communication and in consideration of the huge amount of data on the web, the so-called Web Mining receives considerable interest. The main goal of Web Mining is the mining of information and the analysis of web-based hypertext data on the basis of well-known Data Mining methods. The problem of these Data Mining methods is the discovery of patterns and structures in large amounts of data. Web Mining can be divided into three major fields: Web Structure Mining, Web Content Mining und Web Usage Mining.

The main focus of this thesis is on Web Structure Mining, that is the exploration and examination of structural properties of web-based documents. In this thesis the web will be considered as a hypertext. Graph-theoretic indices for measuring structural characteristics of hypertexts were used in the initial phase of hypertext research. In terms of similarity-based clustering, graph-theoretic indices are inadequate. Thus, the present thesis deals with the development of new graph-theoretic and similarity-based methods for analyzing hypertext structures.

Graph-theoretic methods are only reasonably applicable if they allow a meaningful and efficient structural comparison of hypertexts. Therefore, in the following thesis a parametric graph similarity model with applications in Web Structure Mining will be developed. Developing such a model is challenging because classical methods for measuring the structural similarity of graphs are based on isomorphic relations between the underlying graphs or subgraphs. It is well-known that the subgraph isomorphism problem is NP-complete. In contrast to this, a new method for measuring the structural similarity of graphs will be presented in this thesis that is not based on isomorphic relations.

In the context of analyzing hypertext structures, the vector space model will be used frequently. On the basis of a graph-oriented hypertext representation the hypothesis will be presented and proven, that the graph-oriented representation is a meaningful starting point for modelling web-based hypertexts. On the basis of experimental examinations the developed graph similarity measures will be evaluated successfully. Furthermore, applications of the new graph similarity measures will be presented.





# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation der Arbeit . . . . .	1
1.2 Zielsetzung der Arbeit . . . . .	3
1.3 Aufbau der Arbeit . . . . .	5
1.4 Wissenschaftlicher Beitrag der Arbeit . . . . .	7
<b>2 Strukturelle Aspekte hypertextueller Einheiten</b>	<b>11</b>
2.1 Hypertext und Hypermedia . . . . .	11
2.2 Problemstellungen des Web Mining . . . . .	14
2.2.1 Probleme des World Wide Web bezüglich der Informationssuche . . . . .	14
2.2.2 Bereiche des Web Mining und deren Kernaufgaben . . . . .	16
2.3 Existierende graphentheoretische Analysemethoden von Hypertext- strukturen . . . . .	20
2.3.1 Motivation . . . . .	20
2.3.2 Maße für die strukturelle Analyse von Hypertexten . . . . .	22
2.3.3 Zusammenfassende Bewertung . . . . .	28
2.3.4 Fazit . . . . .	29
2.4 Existierende Clusteringverfahren zur Analyse hypertextueller Daten . . . . .	30
2.4.1 Interpretation von Clusterlösungen . . . . .	32

2.4.2	Hierarchische Clusteringverfahren . . . . .	34
2.4.3	Partitionierende Clusteringverfahren . . . . .	37
2.4.4	Sonstige Clusteringverfahren . . . . .	39
2.5	Modellbildung: Polymorphie und funktionale Äquivalenz . . . . .	41
2.6	Konkreter Modellierungsansatz auf der Basis von GXL . . . . .	44
2.7	Zusammenfassende Bewertung und Fazit . . . . .	46
<b>3</b>	<b>Grenzen der inhaltsbasierten Kategorisierung von Hypertextstrukturen</b>	<b>49</b>
3.1	Motivation . . . . .	49
3.2	Das Testkorpus und die Extraktion web-basierter Hypertexte . . . . .	51
3.3	Motivation des maschinellen Lernverfahrens . . . . .	54
3.4	Charakterisierung des Kategorisierungsexperiments . . . . .	57
3.5	Interpretation der Evaluierungsergebnisse . . . . .	60
3.6	Fazit . . . . .	62
<b>4</b>	<b>Graphentheorie und strukturelle Ähnlichkeit: Bekannte Methoden</b>	<b>65</b>
4.1	Erforderliche Grundlagen . . . . .	65
4.1.1	Überblick und Resultate der Graphentheorie . . . . .	67
4.1.2	Ähnlichkeit strukturierter Objekte . . . . .	70
4.1.3	Abstand, Distanz und Metriken . . . . .	71
4.2	Strukturelle Ähnlichkeit von Graphen . . . . .	73
4.3	Graph Mining und weitere graphorientierte Ähnlichkeitsmaße . . .	78
4.4	Zusammenfassende Bewertung . . . . .	86
<b>5</b>	<b>Graphbasierte Analyse und Retrieval: Neuer Ansatz</b>	<b>91</b>
5.1	Motivation . . . . .	92
5.2	Gradsequenzen von Graphen . . . . .	96

5.3	Hierarchisierte und gerichtete Graphen . . . . .	100
5.4	Zentraler Lösungsansatz . . . . .	103
5.5	Berechnungsgrundlagen . . . . .	106
5.6	Strukturelle Ähnlichkeit hierarchisierter und gerichteter Graphen .	111
5.7	Ergebnisse . . . . .	120
5.8	Experimentelle Ergebnisse . . . . .	122
5.8.1	Experimente mit Website-Strukturen . . . . .	123
5.8.2	Experimente mit web-basierten Dokumenten . . . . .	130
5.8.3	Fazit . . . . .	133
<b>6</b>	<b>Exkurs: Strukturvorhersage</b>	<b>137</b>
6.1	Erkennung struktureller Beziehungen zwischen Graphmengen . . .	137
6.2	Ergebnisse . . . . .	140
6.3	Fazit . . . . .	142
<b>7</b>	<b>Zusammenfassung und Ausblick</b>	<b>143</b>
7.1	Zusammenfassung der Ergebnisse . . . . .	143
7.2	Ausblick . . . . .	146
7.3	Weiterführende Fragestellungen . . . . .	149
	<b>Literaturverzeichnis</b>	<b>151</b>



# Abbildungsverzeichnis

2.1	Crawler-Indexer Architektur auf der Basis der Suchmaschine <b>Alta Vista</b> . . . . .	16
2.2	Der vollständige gerichtete Graph $K_4$ und der entsprechende Graph mit der leeren Kantenmenge. . . . .	22
2.3	Das linke Bild zeigt das Hypertext File <i>abbccbbccd</i> . Jedem Symbol ist das Paar (Level, Ordnung) zugeordnet. Das rechte Bild zeigt einen Hypertext, der aus drei Hypertext Files $e_1$ , $e_2$ und $e_3$ besteht, zusammen mit seiner Linkstruktur. Beispielsweise enthält $e_1$ zwei Matched Pairs, nämlich $x_1$ und $x_2$ . . . . .	28
2.4	Tabellarische Zusammenfassung der Ergebnisse aus Kapitel (2.3.2). . . . .	29
2.5	A: Disjunkte, aber nicht partitionierende Clusterung mit <i>nicht gruppierbaren</i> Objekten. B: Überlappende Clusterung. C: Partitionierende Clusterung . . . . .	31
2.6	Dendrogramm für eine Clusteraufgabe mit acht Dokumenten. Die gestrichelten Linien deuten die gewählten Homogenitätsstufen an. . . . .	35
2.7	Cluster mit ihren Zentren. . . . .	39
2.8	$ D  = 5$ . Der vollständige Graph $G_D$ und sein Teilgraph $G_D^{0.5}$ . . . . .	41
2.9	Schematische Darstellung zweier funktional äquivalenter Präsentationen [156]. Im Fall (A) auf der Basis einer Liste und (B) mit einem <i>compound document</i> [72], welches aus mehreren Webseiten besteht. . . . .	43
2.10	(i): Das linke Bild zeigt einen fiktiven Hypergraph in <b>GXL</b> bestehend aus zwei Webseiten mit internen Graphen und einem Hyperlink. (ii): Das rechte Bild zeigt die graphentheoretische Struktur. . . . .	44
2.11	Ein einfacher gerichteter Hypergraph mit den Hyperkanten $E_1 = (\{v_1, v_2\}, \{v_4, v_5\})$ , $E_2 = (\{v_5, v_3\}, \{v_6\})$ . . . . .	46
2.12	Tabellarische Zusammenfassung der Clusteringverfahren aus Kapitel (2.4). . . . .	47

3.1	Konfigurationsbereich der Einzelextraktion von HyGraph. Im oberen Bereich werden die Grundeinstellungen der Extraktion festgelegt. Der mittlere Bereich der Maske legt die Tokenart und damit die jeweilige „Bag of Words“ fest. Mit Hilfe des unteren Konfigurationsbereichs wird der Extraktionsfilter gesetzt. . . . .	53
3.2	Vereinfachte schematische Darstellung der „Bag of Words“. . . . .	55
3.3	A: Positiv (+1) und negativ (-1) klassifizierte Datenpunkte. $m$ bezeichnet die maximale Trennspanne. Die Supportvektoren, die die Lage der trennenden Hyperebene $E$ maßgeblich bestimmen, sind mit SV markiert. B: Im linken Koordinatensystem ist keine lineare Trennung möglich. Schematische Transformation der Eingabedaten in den höherdimensionalen Featureraum mit anschließender linearer Trennung. . . . .	56
3.4	Kontingenztafel für eine binäre Kategorisierung. . . . .	60
3.5	Ergebnisse der Performancemessung. . . . .	61
3.6	Ergebnisse der Auswertung von $U_i$ . . . . .	61
4.1	Der vollständige Graph $K_5$ mit 5 Knoten und der bipartite Graph $K_{3,3}$ . . . . .	68
4.2	Zwei knotenmarkierte gerichtete Beispielgraphen. . . . .	76
4.3	Zwei knotenmarkierte, geordnete und ungerichtete Wurzelbäume $T_1$ und $T_2$ . Das rechte Bild zeigt ein optimales Alignment von $T_1$ und $T_2$ . . . . .	80
4.4	Ein fiktiver DOM-Tree mit seinen syntaktischen Knotensequenzen: a/b, a/c/e/f, a/c/e/f, a/c/e und a/d. . . . .	85
4.5	Zusammenfassung der Ergebnisse aus den Kapiteln (4.2), (4.3). . . . .	87
4.6	Menge der Isomorphieklassen für ungerichtete Graphen. . . . .	88
5.1	Das web-basierte Graphmatching. . . . .	93
5.2	Wichtige Grundeigenschaft des zukünftigen Graphähnlichkeitsmaßes: Die Bildung der $f(E_i, \bar{E}_i)$ geschieht ebenenweise. . . . .	95
5.3	Ein gerichteter Graph mit der Ordnung $ V  = 4$ . . . . .	97
5.4	Ein 3-regulärer Graph. . . . .	98
5.5	Zwei nicht isomorphe Graphen mit den gleichen Ausgangsgrad- und Eingangsgradsequenzen. . . . .	100

5.6	Kantentypen eines Graphen nach Definition (5.3.1). . . . .	102
5.7	Zwei nicht symmetrische Graphen mit den gleichen Ausgangsgrad- und Eingangsgradsequenzen. . . . .	103
5.8	Induzierte Ausgangsgrad- und Eingangsgradsequenzen auf den Ebenen. . . . .	104
5.9	Der Alignment-Graph $G_{S_1, S_2}$ der Sequenzen $S_1, S_2$ . . . . .	109
5.10	Kerndaten des Graphkorpus $T_C$ . . . . .	123
5.11	Verteilungen der gerankten Ähnlichkeitswerte für $d_1, d_2$ und $d_3$ , bezogen auf alle Datenklassen der Definition (5.8.1). Die $X$ -Achse bezeichnet den Ähnlichkeitswert $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \in [0, 1]$ . Auf der $Y$ -Achse ist die Anzahl der Graphpaare aufgetragen. . . . .	126
5.12	Kummulative Ähnlichkeitswertverteilungen für die Datenklassen der Definition (5.8.1). $X$ -Achse: Ähnlichkeitswert $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \in [0, 1]$ . $Y$ -Achse: Prozentsatz der Graphen, die einen Ähnlichkeitswert $d_3 \leq X$ -Wert besitzen. . . . .	126
5.13	Dendrogramm als Ergebnis der Clusterung von $T_{\text{small}}$ . Die zweiundzwanzig Graphen sind jeweils mit Objektnummern bezeichnet. . .	128
5.14	Web-basierte Hypertextgraphen der Cluster (11,18), (9,16) und (3,7,10). Die Clusterschreibweise in Tupelform verwendet dabei die Objektnummern. Es gilt $\zeta = 0.5$ ; Parameterbelegung siehe Punkt (5) der Definition (5.8.1). . . . .	129
5.15	(i): Das linke Bild zeigt ein einfaches HTML-Codefragment. (ii): Das rechte Bild bildet den entsprechenden DOM-Tree ab. . . . .	130
5.16	Ausgehend von einer Menge von GXL-Repräsentationen erfolgt die web-basierte Extraktion der DOM-Trees mit Hilfe von <code>HTMLParser</code> [257]. Daran anschließend erzeugt <code>HyGraph</code> die jeweilige Ähnlichkeitsmatrix, die als Eingabe des Clusteringverfahrens dient. Die Registerkarten <code>Statistics</code> und <code>Partitions</code> dienen zur Berechnung der Clusterhomogenität und des Abbruchkriteriums aus Kapitel (5.8.1). Dagegen wird mit Hilfe der Registerkarte <code>Visualization of Clustering</code> das entsprechende Dendrogramm erzeugt. . . . .	131
5.17	Bewertung der Clustergüte von $W_1$ , auf der Basis von Precision und Recall. . . . .	132
5.18	Bewertung der Clustergüte von $W_2$ , auf der Basis von Precision und Recall. . . . .	133
5.19	HTML-Repräsentation einer Webseite aus $C_2$ , bezogen auf $W_1$ . . . .	134

5.20	HTML-Repräsentation einer weiteren Webseite aus $C_2$ , bezogen auf $W_1$ . . . . .	134
6.1	Das Schaubild zeigt die gerankten Ähnlichkeitswerte der Graphmengen $C_1$ und $C_2$ . Es gilt $\theta = 0.5$ und $ C_1  =  C_2  = 500$ . $X$ -Achse: Ähnlichkeitswert $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \in [0, 1]$ . $Y$ -Achse: Anzahl der Graphpaare. . . . .	139
6.2	Das Schaubild zeigt die gerankten Ähnlichkeitswerte von $C_1^{\text{new}}$ und $C_2^{\text{new}}$ . Die Auftrennung der gemischten Menge $C = C_1 \cup C_2$ erfolgte mit $\theta = 0.5$ . $X$ -Achse: Ähnlichkeitswert $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \in [0, 1]$ . $Y$ -Achse: Anzahl der Graphpaare. . . . .	140
6.3	Kumulative Ähnlichkeitswertverteilungen von $C_1^{\text{new}}$ , $C_2^{\text{new}}$ und $C$ . Auf der $X$ -Achse ist jeweils der Ähnlichkeitswert $d_3 \in [0, 1]$ aufgetragen. Die $Y$ -Achse bezeichnet den Prozentsatz der Graphen, die einen Ähnlichkeitswert $d_3 \leq X$ -Wert besitzen. . . . .	141
7.1	Schematische Darstellung der ähnlichkeitsbasierten Graphanalyse im Web Usage Mining. . . . .	147



# Kapitel 1

## Einleitung

### 1.1 Motivation der Arbeit

Die Untersuchung von *Strukturen* ist aus der Sicht vieler Wissenschaftsbereiche ein aktuelles Forschungsthema. Dabei ist die Strukturanalyse einerseits in anwendungsorientierten Disziplinen und andererseits in theorieorientierten Forschungsbereichen von zentraler Bedeutung:

- In der Linguistik wird intensiv die Struktur von Sprache, z.B. die *syntaktische Sprachstruktur* [9, 50] untersucht.
- Die soziologische Forschung betrachtet z.B. *Kommunikationsstrukturen* [12] und *soziale Netzwerke* [105, 108, 204].
- In der Biologie und in der Biochemie spielen z.B. *fraktale biologische Strukturen* [206] eine große Rolle.
- Die Elektrotechnik untersucht Strukturen von Stromverzweigungen, elektrischer Netzwerke und Platinen.

Aus diesen Beispielen geht zunächst nicht hervor, mit welchen Methoden und Formalismen die jeweiligen Strukturen modelliert werden.

Da in dieser Arbeit *relationale Strukturen* in Form von *Graphen* als Repräsentation komplexer Dokumentstrukturen eine wesentliche Rolle spielen, ist speziell das letzte Beispiel der obigen Aufzählung interessant. KIRCHOFF [134] publizierte im Bereich der Elektrizitätslehre bereits 1847 eine wichtige Arbeit bezogen auf

die Theorie der Stromverzweigungen, die einen Grundstein der modernen Graphentheorie<sup>1</sup> legte. Daran schlossen sich richtungsweisende Beiträge<sup>2</sup> von CALEY [47], PETERSEN [175] und SYLVESTER [225] an, die ihre Wurzeln ebenfalls in der Graphentheorie besitzen. Heute ist die Beschreibung von Strukturen ohne graphbasierte Modelle in vielen Wissenschafts- und Lebensbereichen nicht mehr vorstellbar, wobei Graphen in der Informatik, z.B. für die Darstellung von Rechnernetzen, breite Anwendung<sup>3</sup> finden.

Die vorliegende Arbeit ist thematisch in einem Teilbereich des *Web Mining* [46, 139] – dem *Web Structure Mining* [139] – angesiedelt, weil sie strukturelle Modellierungsaspekte *web-basierter*<sup>4</sup> Dokumentstrukturen untersucht. Da der Umgang mit Computern allgegenwärtig ist und die Menge an Dokumenten im Web bekanntlich exponentiell zunimmt, sind Hilfsmittel zur schnellen Erfassung, Klassifizierung und Auffindung von Dokumenten von zentraler Bedeutung. Längst wurde klar, dass Inhalt und Struktur vernetzter Dokumente hierbei relevant sind. Die vorliegende Arbeit konzentriert sich auf Strukturaspekte web-basierter Dokumente, welche in jüngerer Zeit immer stärker ins Blickfeld rücken.

Es existieren formale Ansätze [67, 86, 144, 153], die strukturelle Aspekte hypertextueller Dokumente beschreiben. Die ersten bekannten Arbeiten, die insbesondere die strukturelle Analyse von Hypertexten auf der Basis graphentheoretischer Methoden fokussierten, stammen von BOTAFOGO et al. [28, 29, 30]. Dabei wurden bekannte Konzepte<sup>5</sup> der Graphentheorie verwendet, um Maßzahlen – so genannte *Indizes* [63, 155] – für die Beschreibung struktureller Hypertextausprägungen zu entwickeln. Beispielsweise definierten BOTAFOGO et al. [29] als einen typischen Vertreter das bekannte Maß *Compactness*<sup>6</sup>, welches den Grad der *Vernetztheit* einer Hypertextstruktur beschreibt. Die Aussagekraft solcher Maße ist jedoch sehr eingeschränkt, da die zu beschreibende Ausprägung auf eine einzige Maßzahl abgebildet wird. Damit folgt weiter, dass solche Maße nicht eindeutig interpretierbar sind. Unmittelbar daraus resultiert ein Problem, welches sich bislang negativ auf die Analyse hypertextueller Dokumente auswirkte [63]: Wegen der nicht eindeutigen Interpretierbarkeit und der damit verbundenen mangelnden Aussagekraft dieser Maße, ist eine Gruppierung ähnlicher Strukturen nicht möglich, mit dem Ziel, ähnliche Funktionen oder sogar Qualitätsmerkmale abzuleiten. Ein wichtiger Schritt für die Gruppierung strukturell ähnlicher Hypertexte wäre die Entwicklung von Analysemethoden, die ganzheitliche Strukturvergleiche auf zwei

---

<sup>1</sup>Siehe Kapitel (4.1.1).

<sup>2</sup>Weitere historische Beiträge zur Graphentheorie findet man z.B. im ersten Lehrbuch der Graphentheorie, welches von KÖNIG [137] verfasst wurde.

<sup>3</sup>Für weitere Anwendungen siehe Kapitel (4.1.1).

<sup>4</sup>Web ist die Bezeichnung für das *World Wide Web* (WWW) [20].

<sup>5</sup>Siehe Kapitel (2.3.2).

<sup>6</sup>Siehe Kapitel (2.3.2).

gegebenen Hypertextgraphen zulassen.

Strukturelle Vergleiche hypertextueller Graphmuster, bezogen auf die Interpretation lernpsychologischer Fragestellungen, führten z.B. WINNE et al. [247] durch, wobei der Index *Multiplicity*<sup>7</sup> definiert wurde. Dabei ist Multiplicity lediglich auf der Basis der Kantenschnittmenge zweier Graphmuster definiert. Das impliziert, dass signifikante strukturelle Unterschiede zwischen Graphmustern durch die so erhaltenen Ähnlichkeitswerte nicht erfasst werden. Im Hinblick auf eine ähnlichkeitsbasierte Gruppierung folgt schließlich, dass die entstehenden Gruppierungen keine weitreichende Aussagekraft besitzen und damit schlecht interpretiert werden können. Somit scheidet die Klasse von Ähnlichkeitsmaßen, die auf der Basis der Kantenschnittmenge definiert ist, für zukünftige ähnlichkeitsbasierte Analysen aus. Um eine bessere Wirkung hypertextueller Graphvergleiche zu erzielen, welche sich letztlich in einer wesentlich aussagekräftigeren Modellierung web-basierter Hypertexte auswirkt, wird in dieser Arbeit ein deutlich aussagefähigeres Graphähnlichkeitsmodell entwickelt. Die eigentliche Zielsetzung der Arbeit und daraus resultierende Anforderungen werden nun in Kapitel (1.2) dargestellt.

## 1.2 Zielsetzung der Arbeit

In Kapitel (1.1) wurden Probleme graphentheoretischer Indizes kurz gefasst beschrieben. Der Einsatz graphbasierter Repräsentationen zur Modellierung web-basierter Hypertexte im Hinblick auf Anwendungen im Web Structure Mining kann demnach nur dann erfolgreich sein, wenn die darauf aufbauenden Analysemethoden so viel komplexe Strukturmerkmale wie möglich erfassen. Daraus ergibt sich die Anforderung ein Verfahren zu entwickeln, welches die strukturelle Ähnlichkeit graphbasierter Hypertexte ganzheitlich bestimmt. Dies stellt die eigentliche Herausforderung dieser Arbeit dar.

Das Hauptziel dieser Arbeit wird nun folgendermaßen formuliert:

*Das Hauptziel besteht in der Entwicklung ähnlichkeitsbasierter Analysemethoden hypertextueller Dokumente auf der Basis ihrer hierarchischen Graphstruktur, um einerseits anwendungsbezogene Problemstellungen im Web Structure Mining, z.B. die strukturorientierte Filterung, besser als bisher zu lösen. Andererseits sollen die entwickelten ähnlichkeitsbasierten Analysemethoden so flexibel sein, dass sie für graphorientierte Problemstellungen in anderen Forschungsgebieten [73] einzusetzen sind.*

---

<sup>7</sup>Siehe Kapitel (2.3.2).

Die Frage nach der Notwendigkeit eines graphbasierten Repräsentationsmodells für die adäquate Modellierung hypertextueller Dokumente wurde hierbei durch eine grundlegende Arbeit von MEHLER et al. [156] aufgeworfen. Dabei vertreten MEHLER et al. in [156] die These, dass auf Grund der Phänomene *Polymorphie* und *funktionale Äquivalenz* web-basierte Einheiten nicht eindeutig kategorisierbar sind. Da in [156] das bekannte *Vektorraummodell* [81, 153] als Standardrepräsentation für web-basierte Dokumente eingesetzt wurde, ist die Frage nach der Erprobung eines neuen Repräsentationsmodells gerechtfertigt.

In dieser Arbeit wird die These zu Grunde gelegt und belegt, dass die graphbasierte Repräsentation hypertextueller Dokumente einen zentralen Ausgangspunkt einerseits für graphbasierte Modellierungen und ähnlichkeitsbasierte Analysealgorithmen und andererseits für anwendungsorientierte Aufgaben im Web Structure Mining darstellt. Dabei stellt die ganzheitliche Bestimmung der strukturellen Ähnlichkeit graphbasierter Dokumentstrukturen zunächst ein schwieriges Problem dar. Die bekannten Verfahren zur Bestimmung der Graphähnlichkeit beruhen nämlich in vielen Fällen auf Isomorphie- und Untergraphisomorphiebeziehungen [129, 213, 254]. Da diese aus Komplexitätsgründen [5, 234] für Graphen höherer Ordnung nicht anwendbar sind, scheidet diese Verfahrensklasse zur massendaten-orientierten Anwendung im Web Structure Mining aus. Deshalb ist ein Verfahren zur Bestimmung der Graphähnlichkeit im Web Structure Mining nur dann sinnvoll einsetzbar, wenn es große Datenmengen hinsichtlich Graphen höherer Ordnung verarbeiten kann. Eine Vorgehensweise zur Ermittlung geeigneter Verfahren könnte – beispielhaft – sukzessiv folgende Fragen untersuchen:

- Gibt es Ansätze und Ideen, die Isomorphie- und Untergraphisomorphiebeziehungen aus Effizienzgründen umgehen?
- Existieren strukturelle Kennzahlen<sup>8</sup> der zu betrachtenden Graphen, die effizient zu berechnen sind?
- Wenn ja, sind solche Kennzahlen überhaupt zur Definition von Graphähnlichkeitsmaßen aussagekräftig genug?
- Sind ausreichende Möglichkeiten für die Gewichtung unterschiedlicher struktureller Aspekte (z.B. bei hierarchischen Graphen die Berücksichtigung der Höhenunterschiede<sup>9</sup>) gegeben?
- Wie kann weiter vorgegangen werden, falls ein Graphähnlichkeitsmaß gewisse Anforderungen nicht erfüllt? Sind mögliche Defizite auf der Basis von Parametern ausgleichbar?

---

<sup>8</sup>Siehe Kapitel (5.1).

<sup>9</sup>Siehe Kapitel (5.7).

- Ist weitergehend die Entwicklung eines Verfahrens möglich, das auf Grund seiner Konstruktion eine ganze Klasse von Ähnlichkeitsmaßen definiert?
- Sind solche Graphähnlichkeitsmaße nur im Bereich web-basierter Hypertexte nutzbar oder können sie auf Grund ihrer Konzeption überall dort eingesetzt werden, wo Graphähnlichkeitsprobleme bezüglich derselben Graphklasse<sup>10</sup> gestellt werden?

Anhand dieser beispielhaften Vorgehensweise gewinnt man einen Eindruck über die Vielzahl der Fragestellungen, die auf der Suche nach einem Verfahren zur Bestimmung der strukturellen Graphähnlichkeit beantwortet werden müssen. Entsprechend ist es für die vorliegende Arbeit von zentraler Bedeutung ein Graphähnlichkeitsmodell zu entwickeln, welches zur Lösung graphorientierter Problemstellungen im Web Structure Mining und verwandter Aufgaben in anderen Forschungsbereichen beiträgt.

## 1.3 Aufbau der Arbeit

Nach der Einleitung in Kapitel (1) gibt Kapitel (2) einen Überblick über bestehende *Data Mining-Konzepte* [104], wobei vor allem existierende Arbeiten der graphentheoretischen Analyse von Hypertexten detailliert besprochen werden. Weiter werden insbesondere die *Clusteringverfahren* [23, 78] ausführlich diskutiert, da sie in dieser Arbeit ein wichtiges Bindeglied zur ähnlichkeitsbasierten Dokumentanalyse darstellen. Für die Argumentationslinie der Arbeit sind die Phänome Polymorphie [156] und funktionale Äquivalenz [156] von wesentlicher Bedeutung. Vorbereitend für ein Experiment im Bereich der inhaltsbasierten Kategorisierung werden in Kapitel (2) die dazu notwendigen Begriffe, zusammen mit einem graphbasierten Repräsentationsmodell [156], eingeführt.

Das Kapitel (3) zeigt die Grenzen der inhaltsbasierten Kategorisierung in Form eines Experiments auf. Die Hypothese dieses Kapitels ist, dass Polymorphie und funktionale Äquivalenz charakteristisch für web-basierte Einheiten sind. Nach einer formellen Charakterisierung der Problemstellung werden die Ergebnisse der SVM-Kategorisierung<sup>11</sup> interpretiert. Sie untermauern dabei nachhaltig die zu Anfang aufgestellte Hypothese.

Die zusammengefasste Beschreibung des Forschungsstandes und der Kernaufgaben hinsichtlich der Graphentheorie ist Gegenstand von Kapitel (4). Neben

---

<sup>10</sup>Die in dieser Arbeit betrachtete Graphklasse besteht aus knotenmarkierten, hierarchisierten und gerichteten Graphen. Siehe Kapitel (5.3).

<sup>11</sup>Siehe Kapitel (3.3).

einer Diskussion über den Ähnlichkeitsbegriff und der Einführung wesentlicher Begriffe, wie z.B. Metrik, Abstand und Distanz, werden bekannte Methoden zur Bestimmung der strukturellen Ähnlichkeit von Graphen beschrieben. Das Ziel von Kapitel (4) besteht insbesondere darin, die mathematischen Fundamente der existierenden Verfahren zu beleuchten, um damit eine Abgrenzung zum neuen Ansatz leichter zu erreichen.

In Kapitel (5) wird zunächst die Motivation und der zentrale Lösungsansatz zur Bestimmung der Graphähnlichkeit hierarchischer Graphen angegeben. Es stellt sich heraus, dass die Gradsequenzen gerichteter Graphen eine aussagekräftige Basis des neuen Verfahrens darstellen, jedoch nicht in Form einfacher Gradsequenzvektor-Vergleiche<sup>12</sup>. Der wesentliche Aspekt, durch den sich das neue Verfahren von den in dieser Arbeit behandelten bekannten Verfahren abhebt, ist, dass die jeweiligen Graphen zunächst in eindimensionale Strukturen transformiert werden. Die transformierten Strukturen werden auf der Basis bekannter Alignment-Techniken<sup>13</sup> [99] weiterverarbeitet. Ein wichtiger Schritt in dieser Arbeit ist die Anwendung einer Gruppe *multivariater Analyseverfahren* [6], die Clusteringverfahren. Diese tragen zur Lösung anwendungsorientierter Problemstellungen im Bereich des Web Structure Mining bei. Kapitel (5) schließt mit einer experimentellen Untersuchung ab. In dieser werden die entwickelten ähnlichkeitsbasierten Analysemethoden auf bestehende web-basierte Dokumente angewendet.

Während sich der experimentelle Teil aus Kapitel (5) vornehmlich mit der anwendungsbezogenen Interpretation der gewonnenen Clusterlösungen beschäftigt, verfolgt das Kapitel (6) einen darüber hinausgehenden Weg: Anhand vorgegebener Ähnlichkeitswertverteilungen zweier Graphmengen, wird die strukturelle Beziehung zwischen den Graphmengen untersucht. Die Evaluierungsergebnisse belegen, dass das eingesetzte Graphähnlichkeitsmaß zur Erkennung komplexer Graphstrukturen geeignet ist. Weiter untermauern die Ergebnisse dieses Kapitels den sinnvollen Einsatz des verwendeten Graphähnlichkeitsmaßes im Web Structure Mining.

Kapitel (7) fasst die Ergebnisse der Arbeit zusammen. Abschließend erfolgt einerseits ein kurz gefasster Ausblick bezogen auf weitere potenzielle Anwendungsgebiete. Andererseits wird im Rahmen des Ausblicks eine bereits bestehende Anwendung des Graphähnlichkeitsmodells aus Kapitel (5) erläutert, die nicht im Bereich des Web Mining angesiedelt ist, und eine Aufstellung weiterführender Fragestellungen angegeben.

---

<sup>12</sup>Siehe Definition (5.2.2) in Kapitel (5.2).

<sup>13</sup>Siehe Kapitel (5.5).

## 1.4 Wissenschaftlicher Beitrag der Arbeit

Im Bereich der strukturellen Analyse von Hypertexten existieren viele bekannte Arbeiten, z.B. [28, 29, 30, 247, 235], die insbesondere auf graphentheoretischen Modellierungsmethoden basieren. Ein Großteil dieser Arbeiten beschäftigt sich mit der Definition und Analyse graphentheoretischer Indizes, die bereits in Kapitel (1.1) erwähnt wurden. Dabei dienen Indizes meistens zur strukturellen Charakterisierung typischer Hypertextausprägungen und zur Beschreibung von Graphmustern im Zusammenhang mit Hypertext-Navigationsproblemen [151, 152, 235]. Da die Aussagekraft und Interpretierbarkeit solcher Indizes sehr beschränkt ist, eignen sich Indizes nicht für die ähnlichkeitsbasierte Gruppierung von Hypertexten, welche aber den Schlüssel für viele Anwendungen im Web Structure Mining darstellt. Diese Arbeit hat daher den Anspruch, graphentheoretische und ähnlichkeitsbasierte Methoden zur strukturellen Analyse web-basierter Hypertexte zu entwickeln, damit bestehende Analysemethoden erweitert und verbessert werden.

Anstatt des bekannten Vektorraummodells als Standardrepräsentation, wird in dieser Arbeit ein graphbasiertes Repräsentationsmodell erprobt, welches auf hierarchisierten und gerichteten Graphen basiert. Dies geschieht mit dem Ziel, neue Repräsentationsmodelle für eine adäquate Modellierung hypertextueller Dokumente zu erforschen. Die Vorarbeiten für die Entwicklung ähnlichkeitsbasierter Analysemethoden auf der Basis der hierarchischen Graphstruktur erfolgen in Kapitel (3). Kapitel (3) beschäftigt sich mit einem Experiment zur inhaltsbasierten Hypertextkategorisierung. Diesem Experiment liegen die von MEHLER et al. [156] definierten Begriffe Polymorphie und funktionale Äquivalenz zu Grunde, welche hinsichtlich hypertextueller Dokumente neuartig sind.

In Kapitel (5) wird ein zentraler Lösungsansatz zur Bestimmung der strukturellen Ähnlichkeit hierarchisierter und gerichteter Graphen vorgestellt. In der vorliegenden Arbeit findet das Graphähnlichkeitsmodell aus Kapitel (5) Anwendung bezüglich praxisorientierter Problemstellungen im Web Structure Mining. Mit Hilfe des Graphähnlichkeitsmodells wird es möglich, ganzheitliche Strukturvergleiche auf Hypertextgraphen durchzuführen. Im Folgenden werden erzielte Erweiterungen auf der Basis des Graphähnlichkeitsmodells angegeben. Diese Erweiterungen zeigen eine wesentliche Verbesserung des Index-Konzepts auf:

- Auf Grundlage des parametrischen Graphähnlichkeitsmodells ist die Betonung vielfältiger Strukturaspekte möglich, wobei damit alle komplexen Objektausprägungen erfasst werden.
- Im Gegensatz zu Indizes ist nun die Anwendung multivariater Analysemethoden möglich. In dieser Arbeit werden speziell die Clusteringverfahren

gewählt, wobei diese zu den Struktur entdeckenden Verfahren gehören. Auf der Basis aussagekräftiger Graphvergleiche werden damit viele Anwendungen verbessert, z.B. die strukturorientierte Filterung web-basierter Hypertexte.

- Insgesamt erhält man ein generisches Modell zur Messung der strukturellen Ähnlichkeit hierarchisierter und gerichteter Graphen, welches in allen drei Teilbereichen des Web Mining – Web Structure Mining, *Web Usage Mining* und *Web Content Mining* – anwendbar ist. Im Web Usage Mining ist das Graphähnlichkeitsmodell aus Kapitel (5) z.B. zur Erzeugung und Erforschung graphbasierter Benutzergruppen<sup>14</sup> einsetzbar.

Die Bestimmung der strukturellen Ähnlichkeit von Graphen stellt ein mathematisch schweres Problem dar. Klassische Verfahren zur Bestimmung der Graphähnlichkeit beruhen in den meisten Fällen auf Isomorphie- oder Untergraphisomorphiebeziehungen. In Kapitel (4.4) erfolgen eine Diskussion und Bewertung bekannter Verfahren zur Bestimmung der strukturellen Ähnlichkeit von Graphen. Diese zeigen, dass solche Verfahren im Hinblick auf jene graphorientierte Problemstellungen nicht anwendbar sind, bei denen die Verarbeitung von Graphen höherer Ordnung gefragt ist. Eine zentrale Konstruktionsidee des neuen Modells aus Kapitel (5) besteht darin, dass die betrachteten Graphen auf der Basis einer Abbildung in eindimensionale Strukturen transformiert werden. Es stellt sich heraus, dass die Ähnlichkeit der eindimensionalen Strukturen wesentlich effizienter bestimmt werden kann. Aus einer Menge von Ähnlichkeitswerten, die aus Alignments<sup>15</sup> der eindimensionalen Strukturen gewonnen werden, wird schließlich ein finaler Ähnlichkeitswert konstruiert, der die strukturelle Ähnlichkeit zweier Graphen ausdrückt. Kurz gefasst zeichnet sich das neue Modell durch die folgenden Vorteile gegenüber bekannten Verfahren aus:

- Starke Reduktion der Berechnungskomplexität.
- Berücksichtigung komplexer Kantenstrukturen während des Graphvergleichs.
- Hohe Flexibilität durch Parametrisierungsmöglichkeiten.

Auf Grundlage des neuen Modells wurden in dieser Arbeit folgende Ergebnisse erzielt und neue Anwendungsgebiete gefunden:

- Bessere Beschreibungs- und Erforschungsmöglichkeiten bestehender graphbasierter Hypertexte.

---

<sup>14</sup>Siehe Kapitel (7.2).

<sup>15</sup>Siehe Kapitel (5.5).



- Ableitung struktureller Aussagen bezüglich Testkorpora web-basierter Hypertexte. Dies geschieht z.B. auf Grundlage aussagekräftiger Ähnlichkeitswertverteilungen.
- Strukturorientierte Filterung web-basierter Dokumente in Form von DOM-Strukturen. Die Evaluierung des dazugehörigen Clustering-Experiments, welches in Kapitel (5.8.2) durchgeführt wurde, zeichnet sich durch hohe Precision- und Recallwerte aus.
- Das Graphähnlichkeitsmodell aus Kapitel (5) wurde von EMMERT-STREIB et al. [73] verwendet, um eine effiziente Methode zur Klassifikation großer ungerichteter Graphen zu entwickeln. Die binäre Graphklassifikationsmethode wurde u.a. erfolgreich auf *Microarray-Daten* [42] aus Gebärmutterhalskrebs-Experimenten angewendet, mit dem Ziel, Tumorstadien zu unterscheiden [73].



# Kapitel 2

## Strukturelle Aspekte hypertextueller Einheiten

Die Anwendung von klassischen Data Mining-Konzepten [104] auf web-basierte Daten, wie z.B. die Clusteranalyse, wird als Web Mining [46] bezeichnet. Ein Teilbereich des Web Mining, der in dieser Arbeit besonders im Vordergrund steht, ist das Web Structure Mining, welches die Aufdeckung und die Erforschung struktureller Aspekte web-basierter Hypertexte zum Hauptziel hat. Ausgehend von einer kurzen Darstellung der Grundlagen von Hypertext und Hypermedia in Kapitel (2.1) hat das vorliegende Kapitel (2) das Ziel, eine verständliche Einführung von Data Mining-Konzepten im Hinblick auf die Anwendung im Web Mining zu geben. Das Teilgebiet Web Structure Mining wird dabei besonders hervorgehoben, insbesondere graphentheoretische Methoden zur strukturellen Analyse von Hypertexten.

### 2.1 Hypertext und Hypermedia

Bekanntlich ist beim klassischen Medium *Buch* die Struktur und in der Regel auch die Lesereihenfolge sequenziell. Dagegen ist die Kerneigenschaft von *Hypertext*<sup>1</sup>, dass die textuellen Informationseinheiten, die so genannten *Knoten*, auf der Basis von *Verweisen*, auch *Links* genannt, in Form eines gerichteten Graphen, also *nicht linear*, miteinander verknüpft sind [140]. Die einfachste graphentheoretische Modellierung einer Hypertextstruktur ist die Darstellung als unmarkierter gerichteter Graph  $\mathcal{H} := (V, E)$ ,  $E \subseteq V \times V$ .  $V$  heißt Knotenmenge und  $E$  heißt Kantenmenge. Weiter bezeichnet man ein Element  $v \in V$  als Knoten und  $e \in E$

---

<sup>1</sup>In dieser Arbeit bezeichnet ein „Hypertext“ konkrete Ausprägungen oder Instanzen (vgl. im Web: eine „Website“); Hypertext subsummiert in der vorliegenden Arbeit „Hypermedia“. Software zur Handhabung von Hypertexten sei als „*Hypertextsystem*“ bezeichnet.

als gerichtete Kante. Der Hypertext-Begriff wird in den Geisteswissenschaften und der modernen Informatik unterschiedlich interpretiert [238]. So kann man abhängig von der Fachdisziplin und vom Autor durchaus auf unterschiedliche Definitionen des Hypertextbegriffs stoßen. Hypertext wird somit oft als Technologie, Methode oder Metapher bezeichnet [238]. Tatsächlich wurden in der Literatur unzählige Definitionen und Ausprägungen von Hypertext gegeben, siehe z.B. [48, 54, 64, 100, 165, 171, 210]. Bei dieser Fülle von Definitionen – wobei die Autoren unterschiedliche Aspekte herausstellen – betont HOFMANN [113] vier wichtige Kernpunkte, die er für eine vollständige Charakterisierung von Hypertext in der Informatik als notwendig ansieht:

- Hypertexte haben die Gestalt von gerichteten Graphen (Netzwerke). Die Knoten enthalten bzw. repräsentieren die Informationen, die durch Verweise, die Links, miteinander verknüpft sind.
- Sowohl das Lesen als auch das Schreiben von Hypertext sind nichtlineare Tätigkeiten. Eine Datenstruktur, die diese Vernetzung unterstützt, ist dabei die Voraussetzung.
- Hypertexte sind nur in einem *medialen* Kontext, also maschinenunterstützt denkbar. Direkte Anwendungen davon sind klassische Hypertext- und Online-systeme.
- Hypertexte besitzen einen *visuellen* Aspekt. Das bedeutet, dass Hypertext nicht nur ein Konzept der Informationsstrukturierung, sondern auch eine Darstellungs- und Zugriffsform von textuellen Informationen ist.

Auch in der Sprachwissenschaft und in der Linguistik wurde Hypertext als eine neue Form der schriftlichen Sprachverwendung studiert, z.B. [150, 222]. Dabei wurden insbesondere linguistische Aspekte, wie *Kohärenz-* und *Kohäsionsbeziehungen*, in Hypertext untersucht. Eine bekannte Studie in diesem Problemkreis wurde von STORRER [221] durchgeführt. In dieser Arbeit geht es im Wesentlichen um die Fragestellung, ob die Ergebnisse über Untersuchungen von Kohärenzbildungsprozessen in linear organisierten Texten auf den Entwurf von Hypertexten übertragbar sind. Weiterhin wurde die Problemstellung der *automatischen Generierung* von Hypertext aus natürlichsprachigem Text untersucht, insbesondere wie und unter welchen Kriterien Hypertext automatisiert konstruierbar ist. Ein linguistisches Kriterium, welches als Grundlage zur Generierung von Hypertext aus Texten dient, wurde von MEHLER [153] angegeben.

Historisch gesehen wurde die Hypertext-Idee aus heutiger Sicht zweifellos von BUSH [40] geschaffen. In seinem bekannten Artikel „As we may think“ [40] beschrieb er das System **Memex** (Memory Extender), welches zum Ziel hatte, wissenschaftliche Dokumente nichtlinear zu verknüpfen und zu speichern, um dadurch

die schon damals ständig wachsende Anzahl an wissenschaftlichen Publikationen für ein breites Publikum nutzbar zu machen. In seiner Ganzheit wurde dieses System jedoch nie realisiert, zumal es inkompatible Technologien (z.B. Buch und Microfiche) hätte überbrücken müssen. Der eigentliche „Hypertext“-Begriff wurde in den sechziger Jahren durch NELSON geprägt. Er führte die Ideen BUSH's weiter, indem er die technischen Voraussetzungen schaffte, um Hypertext auf Computersystemen zu realisieren. NELSON gilt als Architekt des universellen Hypertextsystems **Xanadu** [164], das aber oft als unrealistisch angesehen wurde, da es zum Ziel hatte, die Gesamtheit aller elektronischen Publikationen weltweit zu integrieren. Die Implementierung von **Xanadu** ist nur in Teilen erfolgt und wird bis heute fortgesetzt [166]. Ein weiterhin sehr bekanntes Hypertextsystem ist **Augment** [74], welches 1962 bis 1976 von ENGLEBART in Stanford realisiert wurde. Insgesamt gesehen wurden viele Hypertextsysteme entwickelt, wobei bekannte Vertreter z.B. **HyperCard**, **NoteCards**, **Neptune/HAM** und **HyperTies** [195, 219] sind. Detaillierte Informationen bezüglich der genannten Hypertextsysteme findet man in [113, 195, 219].

Der Begriff Hypermedia wird üblicherweise gebraucht, wenn in *Hypermedia-Dokumenten*<sup>2</sup> nicht nur Texte, sondern auch multimediale Objekte wie Graphiken, Ton- und Filmsequenzen nichtlinear miteinander verknüpft werden. In der Literatur wird auf Grund dieses Sachverhalts bisweilen Hypertext (textbasiert) und Hypermedia (medienbasiert) als zwei disjunkte Kategorien betrachtet. Für diese Arbeit ist es sinnvoller Hypermedia unter Hypertext zu subsummieren. Hypertext beschreibt dann Dokumente mit Graphstruktur, Hypermedia meint die Untermenge, welche mehrere Medien einbezieht. *Multimediasysteme* werden in der Literatur klar von Hypertextsystemen unterschieden [113, 219], da in Multimediasystemen die Dokumentstrukturen modelliert werden, ohne deren *strukturelle* Aspekte hervorzuheben. Tiefere Einblicke über Hypermedia- und Multimediasysteme geben z.B. STEINMETZ [220] und SCHULMEISTER [201], wobei SCHULMEISTER insbesondere didaktische und lernbezogene Aspekte von Hypermedia behandelt.

Als Anwendungsgebiete von Hypertext und Hypermedia kommen mittlerweile unterschiedlichste Wissenschafts- und Industriebereiche in Frage. Anwendungsgebiete sind beispielsweise *Büro und Management*, *Konstruktions- und Fertigungsbereiche*, *Schule und Weiterbildung*, *technische Dokumentenverwaltung*, *elektronische Enzyklopädien und Bücher*, *hypertextuelle Produktkataloge* und die *Wissensrepräsentation* [138, 195, 235]. Weitere Überblicke über die unterschiedlichen Anwendungsfelder sind in [167, 219, 220] zu finden.

---

<sup>2</sup>Im Sprachgebrauch ist „Hypertext“ wie bereits definiert gebräuchlich, aber nicht „Hypermedia“, sondern „Hypermedia-Dokumente“.

## 2.2 Problemstellungen des Web Mining

Durch die Entstehung des World Wide Web [20], auch Web oder kurz WWW genannt, ist die Popularität von Hypertext in den neunziger Jahren deutlich gestiegen. 1989 wurde von BERNERS-LEE, einem damaligen Mitarbeiter des Forschungszentrums für Teilchenphysik (CERN) in Genf/Schweiz, die Idee des World Wide Web als Hypertextsystem geboren [19].

Da in der vorliegenden Arbeit die Entwicklung graphentheoretischer Modelle für web-basierte Dokumentstrukturen fokussiert wird, erfolgt zunächst ein kurzer Überblick über die Eigenschaften und Probleme des World Wide Web hinsichtlich der Informationssuche. Weiterhin werden die Kernbereiche des Web Mining detailliert dargestellt, wobei in dieser Arbeit das Web Structure Mining besonders thematisiert wird. Dies geschieht vor dem Hintergrund, dass das graph-basierte Modell aus Kapitel (5) zur Berechnung der strukturellen Ähnlichkeit web-basierter Hypertexte, zur Lösung von Problemstellungen im Web Structure Mining beiträgt.

### 2.2.1 Probleme des World Wide Web bezüglich der Informationssuche

Im klassischen Information Retrieval (IR) [7, 81] werden auf der Basis von Informationssystemen Fragestellungen der inhaltsorientierten Auffindung und Gewinnung (Retrieval) von Informationen in großen Datenbeständen untersucht. Dabei ist eine Benutzeranfrage an das System von zwei im Information Retrieval enthaltenen wesentlichen Begriffen geprägt [7, 81, 192]:

- *Vagheit*: Das Informationsbedürfnis kann durch den Benutzer nicht präzise und formal formuliert werden.
- *Unsicherheit*: Sie wird meistens durch die nicht aussagekräftige *Semantik*, also durch fehlende inhaltliche Informationen in den vorliegenden Dokumenten oder Texten induziert.

Vereinfacht gesehen, kann man das World Wide Web als sehr große und inhomogene Datenbank betrachten, die täglich viele Millionen Benutzeranfragen über die verfügbaren Suchdienste erhält. BAEZA-YATES et al. [7] stellen die Probleme des World Wide Web hinsichtlich der Informationssuche detailliert vor: Einerseits bezüglich der Daten und andererseits bezogen auf systemabhängige Benutzeranfragen und deren Interpretation. Der erstgenannte Problemkreis wird dabei in folgende Unterpunkte untergliedert:

- *Verteilte Daten*: Die Daten sind auf Grund der netzwerkartigen Struktur des Webs auf viele Plattformen verteilt, wobei die Rechner in unbekannter Weise miteinander vernetzt sind und ihre Funktionssicherheit stark variiert.
- *Hoher Anteil an unbeständigen Daten*: Große Datenmengen ändern sich innerhalb kurzer Zeit. 1999 wurde ermittelt, dass sich zu dieser Zeit ca. 40% vom Gesamtinhalt des World Wide Web monatlich änderte.
- *Große Datenmengen*: Das Web unterliegt einem exponentialen Datenwachstum, das Skalierungsprobleme induziert.
- *Unstrukturiertheit und Redundanz*: Die meisten Dokumente im Web sind unstrukturiert und inkonsistent, insbesondere HTML-Seiten. Große Datenmengen werden kopiert oder gespiegelt, wodurch beachtliche Mengen an redundanten Daten entstehen.
- *Qualität der Daten*: Da es eine unzureichende *Datenkontrolle* gibt, die z.B. inhaltlich fehlerhafte Dokumente im World Wide Web vor dem *Upload* filtert, kann jeder beliebige Benutzer Daten einstellen, was die Qualität der Ergebnisse von Suchanfragen sehr beeinträchtigt.
- *Heterogenität der Daten*: Die Daten besitzen unterschiedliche *Datentypen*, z.B. Text, Graphik und Video und unterschiedliche Sprachalphabete.

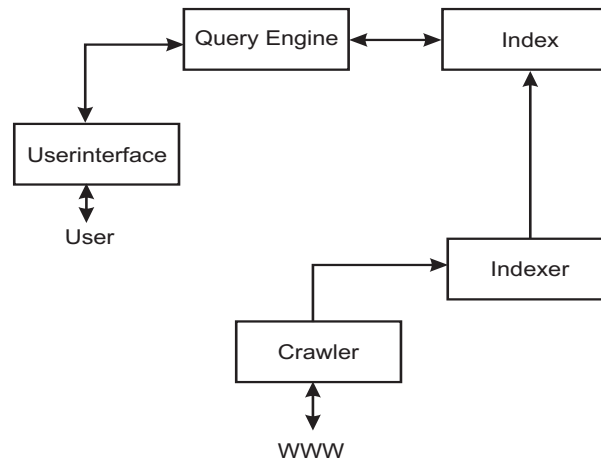
Der zweite Problemkreis umfasst im Wesentlichen die Kernpunkte:

- Richtige Formulierung von Benutzeranfragen und deren Interpretierbarkeit.
- Interpretation von Systemantworten – u.a. die Selektion von „nutzbaren“ Treffern – und Umgang/Optimierung von großen Trefferlisten.

Auf Grund der aufgeführten Probleme wird klar, dass das Ziel, brauchbare Benutzeranfragen zu formulieren und Systemantworten auf der Basis von Information Retrieval-Methoden zu optimieren, eine große Herausforderung darstellt. Um eine bessere Vorstellung von den Komponenten einer Suchmaschine zu bekommen, sei die Abbildung (2.1) [7] betrachtet. Am Beispiel dieser Abbildung werden die wesentlichen Komponenten der Suchmaschine kurz umrissen, die hier aus zwei Blöcken bestehen: (i) aus dem *Benutzer-Interface* und der so genannten *Query Engine* und (ii) aus dem *Crawler* und dem *Indexer*. Wenn die Anfrage über das Benutzer-Interface zur Query Engine übertragen wird, führt die Query Engine eine Datenbankabfrage aus, mit dem Ziel, eine Rangordnung der Ergebnisdokumente zu erzeugen. Die Güte solcher Abfragen wird oft mit den Performancemaßen [81] *Recall* und *Precision*<sup>3</sup>, die aus dem Information Retrieval stammen,

---

<sup>3</sup>Die Definitionen von Recall und Precision werden in Kapitel (3.5) auf der Basis einer Kontingenztafel angegeben.



**Abbildung 2.1:** Crawler-Indexer Architektur auf der Basis der Suchmaschine Alta Vista

gemessen. Der Indexer bestimmt dabei, welche Inhaltsfragmente zur Indexierung gewählt werden, z.B. *Plaintext*, *Ankertexte* oder *Meta-Tags*. Das Sammeln der web-basierten Dokumente übernimmt der Crawler, wobei die *Breiten-* und *Tiefensuche* bekannte Suchstrategien von Crawlern sind. Detaillierte Ausführungen über die Hintergründe von Suchstrategien im World Wide Web sind bei CHAKRA-BARTI [46] und BAEZA-YATES et al. [7] zu finden. Eine umfassende Darstellung der Infomationssuche im World Wide Web mit Hinweisen zur Optimierung von Benutzeranfragen an Suchmaschinen liefert GLÖGGLER [96].

### 2.2.2 Bereiche des Web Mining und deren Kernaufgaben

In der wissenschaftlichen Literatur werden die Begriffe „Data Mining“ und „Wissensentdeckung“ oft unterschiedlich definiert [21, 80]. So geben z.B. WROBEL et al. [250] die Definition des Begriffs „Wissensentdeckung“ folgendermaßen an [80, 250]:

*„Wissensentdeckung in Datenbanken ist der nichttriviale Prozess der Identifikation gültiger, neuer, potenziell nützlicher und schlussendlich verständlicher Muster in (großen) Datenbeständen.“*

Als Teilschritt des Wissensentdeckungs-Prozesses bezeichnen WROBEL et al. [250] Data Mining als den eigentlichen Analyseschritt, das heißt, die Suche und Bewertung von Hypothesen. Entsprechend werden in kommerziellen<sup>4</sup> Bereichen Data Mining-Verfahren [22, 104, 249] oft eingesetzt, um die gigantischen Datenmengen

<sup>4</sup>Wissensentdeckung und Data-Mining werden im kommerziellen Bereich meistens nicht unterschieden [250].



in vielen industriellen und wissenschaftlichen Bereichen zu analysieren und dabei neues Wissen zu generieren. Beispielsweise liegen in vielen Unternehmen große Mengen von Kundendaten vor, jedoch ist das Wissen über die Anforderungen und über das Verhalten der Kunden oft unzureichend. Solche Datenbestände werden in *Data Warehousing*-Systemen gespeichert und mit Methoden des Data Mining untersucht. Das Ziel einer solchen Untersuchung ist die Entdeckung von statistischen Besonderheiten und Regeln innerhalb der Daten, die beispielsweise für Studien des Kunden- oder Kaufverhaltens eingesetzt werden. Die Schwerpunkte der Data Mining-Methoden, die oft in der Praxis angewendet werden, lassen sich mit Hilfe der folgenden Übersicht erläutern:

- Die Suche nach *Assoziationsregeln* [109]: Ein bekanntes Beispiel ist die so genannte *Warenkorbanalyse*, die zum Ziel hat, aus dem aktuellen Kaufverhalten Assoziationsregeln für zukünftiges Kaufverhalten abzuleiten.
- Die *Clusteranalyse* [78]: Der entscheidende Unterschied zwischen der Clusteranalyse und der *Kategorisierung* ist, dass bei der Clusteranalyse das Klassensystem von vornherein unbekannt ist. Das Ziel ist die Gruppierung<sup>5</sup> der Datenobjekte in Gruppen (Cluster), so dass sich die Objekte innerhalb eines Clusters möglichst ähnlich und zwischen den Clustern möglichst unähnlich sind. Dabei basiert die Ähnlichkeit zwischen den Objekten auf einem jeweils problemspezifischen Ähnlichkeitsmaß.
- Die *Kategorisierung* [68]: Sie stellt Verfahren für die Einordnung von Objekten in Kategoriensysteme bereit. Die Kategorisierung stellt mit Hilfe von Zusammenhängen zwischen gemeinsamen Mustern und Merkmalen ein Kategoriensystem für die vorhandenen Objekte her, um dann auf der Basis eines statistischen Kategorisierungsmodells unbekannte Objekte in das Kategoriensystem einzuordnen. Bekannte Kategorisierungsverfahren stammen dabei aus dem Bereich des *Maschinellen Lernens* [109].
- Die *Regressionsanalyse* [109]: Die Regressionsanalyse ist ein Verfahren aus der mathematischen Statistik, welches auf Grund von gegebenen Daten einen mathematischen Zusammenhang in Gestalt einer Funktion zwischen zwei oder mehreren Merkmalen herstellt. Ein bekanntes Beispiel ist die *lineare Regression* [109].

Durch die äußerst starke Entwicklung des World Wide Web gewinnt die Anwendung von Data Mining-Verfahren auf web-basierte Daten immer mehr an Bedeutung. Während das Allgemeinziel des Web Mining die Informationsgewinnung und die Analyse der Webdaten ist, werden drei bekannte Teilbereiche detailliert unterschieden [53, 139, 178, 217]:

---

<sup>5</sup>Die Gruppierung wird in dieser Arbeit auch als Clusterung bezeichnet.

- *Web Content Mining*: Das World Wide Web enthält mittlerweile viele Milliarden von Webseiten, täglich kommen hunderttausende dazu. Das Web Content Mining stellt Methoden und Verfahren bereit, mit deren Hilfe Informationen und damit neues Wissen aus dieser Datenflut automatisch extrahiert werden können. Diese Verfahren finden beispielsweise bei der Informationssuche mit *Suchmaschinen* im World Wide Web Anwendung. Während bekannte Suchmaschinen, wie z.B. *Yahoo*, auf einer einfachen textuellen Schlagwortsuche basieren, stellt die Konzeption neuer, besserer Verfahren für die Informationssuche im Bereich des Web Content Mining immer noch eine große Herausforderung dar. Die aktuellen Suchmaschinen sind nämlich kaum in der Lage, *semantische Zusammenhänge* zwischen web-basierten Dokumenten zu detektieren bzw. die Dokumente nach semantischen Gesichtspunkten zu kategorisieren.
- *Web Structure Mining*: Die Aufgabe des Web Structure Mining ist es, strukturelle Informationen von Websites zu nutzen, um inhaltliche Informationen zu gewinnen, wobei die *interne und externe Linkstruktur* dabei eine wichtige Rolle spielt. Interne Linkstrukturen können mit Auszeichnungssprachen wie HTML oder XML abgebildet werden und beschreiben innerhalb eines Knotens eingebettete graphentheoretische Strukturen. Die externe Linkstruktur beschreibt die Verlinkung der Webseiten untereinander und lässt sich in Form eines *hierarchisierten und gerichteten Graphen* darstellen. Die Graphstruktur des World Wide Web wurde in den letzten Jahren in vielen Arbeiten intensiv untersucht [1, 65, 141, 177], wobei diese Studien zur Entwicklung und Verbesserung von Suchalgorithmen im World Wide Web führten [31, 43, 135, 216]. Weiterhin wurden *Ausgangsgrad-* und *Eingangsgradverteilungen* [65] von Knoten, *Zusammenhangskomponenten* [65] und der *Durchmesser* [65] des WWW-Graphen untersucht. Detaillierte Ergebnisse solcher Untersuchungen sind z.B. in [32, 65, 117, 141, 142, 177, 242, 243] zu finden. Eine der bekanntesten Arbeiten, die im Bereich des Web Structure Mining eine wichtige Anwendung innerhalb der bekannten Suchmaschine *Google* gefunden hat, stammt von KLEINBERG [135]. Dabei führte er die Begriffe *Hubs* und *Authorities* ein. KLEINBERG bezeichnet Authorities als Webseiten, die aktuelle und „inhaltlich brauchbare“ Informationen enthalten, wobei sich diese graphentheoretisch durch hohe Knoten-Eingangsgrade auszeichnen. Dagegen werden Hubs als solche Webseiten bezeichnet, die viele „nützliche Links“ zu gewissen Themengebieten offerieren. Ein guter graphentheoretischer Indikator für potenzielle Hubs ist nach KLEINBERG ein hoher Knoten-Ausgangsgrad der betrachteten Webseite.
- *Web Usage Mining*: Unter dem Web Usage Mining [178] versteht man die Suche und Analyse von Mustern, die auf das Nutzungsverhalten eines WWW-Benutzers schließen lässt. Üblich ist dabei die Anwendung von

Data Mining-Verfahren mit dem Ziel, das Zugriffsverhalten mit Hilfe von *Web-Logs* zu protokollieren. Die Ergebnisse solcher Analysen sind für Unternehmen, besonders aber für Online-Versandhäuser aller Art interessant, weil aus ihnen Aussagen zur Effektivität, zur Qualität und zum Optimierungsbedarf der Websites abgeleitet werden können. Da bei vielbesuchten Websites täglich große Datenmengen von Web-Logs anfallen, kann der Einsatz von Data Warehouse-Systemen notwendig werden, um diese Datenmengen zielgerecht und effizient zu verarbeiten.

Die Bedeutung und Vertiefung des für diese Arbeit relevanten Web Structure Mining soll hier anhand von zwei weiteren Problemstellungen hervorgehoben werden, und zwar im Wesentlichen als Motivation für die weiteren Kapitel:

1. Das Allgemeinziel des Web Structure Mining ist die Erforschung der strukturellen Eigenschaften von web-basierten Dokumentstrukturen und den daraus resultierenden Informationen. An diesem Ziel orientierend, soll hier auf ein Problem aufmerksam gemacht werden, das bei der inhaltsorientierten Kategorisierung von web-basierten Hypertexten auftritt. MEHLER et al. [156] stellten die Hypothese auf, dass die beiden Phänomene funktionale Äquivalenz und Polymorphie charakteristisch für web-basierte Hypertextstrukturen sind. Dabei bezieht sich der Begriff der funktionalen Äquivalenz auf das Phänomen, dass dieselbe Funktions- oder Inhaltskategorie durch völlig verschiedene Bausteine web-basierter Dokumente manifestiert werden kann. Der Begriff der Polymorphie bezieht sich auf das Phänomen, dass dasselbe Dokument zugleich mehrere Funktions- oder Inhaltskategorien manifestieren kann. Dabei werden die Problemstellung und die neuen Begriffe in Kapitel (2.5) definiert. Das Kategorisierungsexperiment, das die oben genannte Hypothese untermauert, wird in Kapitel (3.4) charakterisiert.
2. Im Hinblick auf die Bestimmung der Ähnlichkeit web-basierter Hypertexte fassen *Dokument Retrieval*-Anwendungen die Dokumentstrukturen als die Mengen ihrer Wörter auf und berechnen auf der Basis des Vektorraummodells deren Ähnlichkeit. Als Motivation für graphorientierte Problemstellungen im Web Structure Mining und für die Kapitel (2.4), (5), wird an dieser Stelle ein Verfahren zur Bestimmung der strukturellen Ähnlichkeit web-basierter Dokumente erwähnt, das nicht auf der vektorraumbasierten Repräsentation beruht, sondern auf der Graphdarstellung der hypertextuellen Dokumente. Ausgehend von der automatisierten Extraktion der Hypertexte und einer GXL-Modellierung [248] der Graphen, werden hierarchisierte und gerichtete Graphen erzeugt, die komplexe Linkstrukturen berücksichtigen [156]. Basierend auf diesen Graphrepräsentationen wird in Kapitel (5) das neue Verfahren [62, 73] zur Bestimmung der strukturellen Ähnlichkeit

solcher Graphen entwickelt. Die für das Web Structure Mining resultierenden Anwendungsgebiete werden als Motivation für das neue Verfahren in Kapitel (5.1) dargestellt.

## 2.3 Existierende graphentheoretische Analysemethoden von Hypertextstrukturen

Wie in Kapitel (2.1) bereits dargestellt, lässt sich die auszeichnende strukturelle Eigenschaft von Hypertext, die Nichtlinearität, in Form eines Netzwerks mit Hilfe einer graphentheoretischen Modellierung beschreiben. Damit liegt die Frage nach der Einsetzbarkeit von graphentheoretischen Analysemethoden auf der Hand. Das vorliegende Kapitel (2.3) fokussiert die Realisierbarkeit graphbasierter Modellierungen und gibt einen Eindruck über die Tragfähigkeit der Aussagen, die man mit einfachen graphentheoretischen Modellen, angewendet auf die Hypertextstruktur, erzielen kann.

### 2.3.1 Motivation

Als erste Anwendung für graphorientierte Methoden sei die Analyse des „*Lost in Hyperspace*“-Problems [69, 235] genannt. Aus der Natur der graphbasierten Modellierung, einer hohen Komplexität der vorliegenden Hypertextstruktur, einem fehlenden kontextuellen Zusammenhang der Links und der Tatsache, dass der Navigierende nur einen eingeschränkten Bereich im Hypertextgraph rezipiert, folgt, dass der Hypertextbenutzer die Orientierung verlieren kann. Graphentheoretische Analysemethoden, die als Abstraktionswerkzeug zu verstehen sind, werden oft eingesetzt, um das „Lost in Hyperspace“-Problem besser unter Kontrolle zu halten. Dazu werden graphentheoretische Kenngrößen definiert, die beispielsweise Aussagen über die Erreichbarkeit von Knoten und deren Einfluss im Hypertextgraph treffen [28, 29, 69]. Die Definition von Indizes [63, 155] zur Beschreibung typischer Ausprägungen von Hypertextgraphen kann als weitere Motivation für den Einsatz graphbasierter Methoden angesehen werden. Beispielsweise können solche Maße von *Hypertextautoren* eingesetzt werden, um den Vernetztheitsgrad und die *Linearität* einer Hypertextstruktur zu bestimmen [29]. Eine weitaus tiefer gehende Fragestellung wäre an dieser Stelle, ob man auf der Basis von graphentheoretischen Indizes eine Gruppierung von ähnlichen Strukturen vornehmen könnte, um dann auf ähnliche Funktionen und Qualitätsmerkmale zu schließen. In jedem Fall müssen aber Fragen nach der *Einsetzbarkeit* und der *Interpretierbarkeit* solcher Maßzahlen gestellt werden, die in Kapitel (2.3.3) kurz diskutiert werden.

Das Kapitel (2.3.2) gibt im Wesentlichen einen Überblick über die bekannten Ar-

beiten der graphentheoretischen Analyse von Hypertextstrukturen, wobei es nicht den Anspruch auf Vollständigkeit erhebt. Einerseits werden damit Möglichkeiten vorgestellt, wie man mit einfachen graphbasierten Mitteln Hypertexte auf Grund charakteristischer Eigenschaften beschreiben und solche Maße auf Probleme der Hypertextnavigation anwenden kann. Andererseits zeigen einige der nachfolgenden Arbeiten die Grenzen von graphentheoretischen Maßzahlen auf, die sich z.B. in der Allgemeingültigkeit ihrer Aussagekraft und in der Interpretierbarkeit ihrer Wertebereiche äußern.

Abgesehen von der graphentheoretischen Analyse von Hypertextstrukturen, besteht nach Meinung vieler Autoren im Hypertextumfeld ein deutlicher Mangel an grundlegenden formalen Konzepten, um komplexere hypertextuelle Strukturmerkmale, wie beispielsweise die semantische und pragmatische Unterscheidung von Knoten und Links, mit mathematischen Modellen auszudrücken, siehe z.B. TOCHTERMANN et al. [230]. Dennoch gibt es viele Arbeiten, in denen verschiedenartige Aspekte von Hypertext und Hypertextsystemen formalisiert wurden, z.B. [67, 85, 86, 144, 159, 153, 174, 223], die aber oft nur spezielle Fälle oder Modellierungsaspekte adressieren.

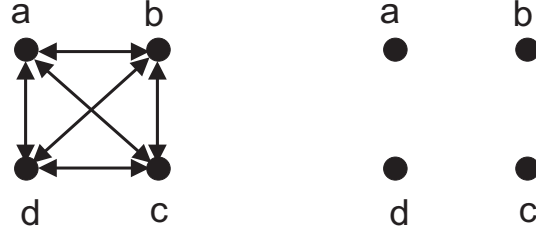
Ein Teilgebiet der strukturellen Analyse von Hypertexten ist speziell die Untersuchung von Hypertextstrukturen mit graphentheoretischen Methoden. Dabei werden die Hypertexte oft in Matrixstrukturen abgebildet, meistens mit dem Ziel, Maßzahlen zu bilden, die zur strukturellen Charakterisierung oder zur Beschreibung von Graphmustern dienen. Die in der Fachliteratur existierenden Ansätze und Arbeiten, die sich mit der graphentheoretischen Analyse und Beschreibung von Hypertextstrukturen beschäftigen, verfolgen im Wesentlichen zwei Ziele:

- Die strukturelle Beschreibung und Charakterisierung von Hypertexten durch *globale* graphentheoretische Maße<sup>6</sup>. Sie heißen global, weil sie auf der gesamten Hypertextstruktur definiert sind und die Hypertexte ganzheitlich charakterisieren. Bekannte Beispiele sind die Hypertextmetriken *Compactness* und *Stratum* von BOTAFOGO et al. [29].
- Die Suche, die Bestimmung und die graphentheoretische Interpretation von Graphmustern in Hypertexten: Solche spezifischen Graphmuster werden oft bei der Beschreibung von Hypertext-Navigationsproblemen [151, 152, 235] und im Zusammenhang von Lernproblemen [168, 179, 247] mit Hypertext analysiert und interpretiert.

In Kapitel (2.3.2) werden nun bekannte Arbeiten vorgestellt, die einerseits die Definition graphentheoretischer Indizes und andererseits die Untersuchung von Hypertext-Navigationsproblemen thematisieren.

---

<sup>6</sup>Solche graphentheoretischen Maße heißen auch Indizes [63, 155].



**Abbildung 2.2:** Der vollständige gerichtete Graph  $K_4$  und der entsprechende Graph mit der leeren Kantenmenge.

### 2.3.2 Maße für die strukturelle Analyse von Hypertexten

Die ersten einschneidenden Arbeiten im Bereich der strukturellen Analyse stammen von BOTAFOGO et al. [28, 29, 30]. In [29] wurden die bekannten Hypertextmetriken Compactness und Stratum definiert, wobei in dieser Untersuchung Hypertextgraphen als unmarkierte gerichtete Graphen  $\mathcal{H} = (V, E)$ ,  $E \subseteq V \times V$  aufgefasst werden. Mit Hilfe der *konvertierten Distanzmatrix*

$$(\mathcal{KDM}_{ij})_{ij} := \begin{cases} w_{ij} & : \text{falls } w_{ij} \text{ existiert} \\ \mathcal{K} & : \text{sonst,} \end{cases} \quad (2.1)$$

wobei  $w_{ij}$  den kürzesten Weg<sup>7</sup> von  $v_i$  nach  $v_j$  und  $\mathcal{K}$  die Konvertierungskonstante<sup>8</sup> bezeichnet, wird Compactness definiert als

$$\mathcal{C} := \frac{(|V|^2 - |V|) \cdot \mathcal{K} - \sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \mathcal{KDM}_{ij}}{(|V|^2 - |V|) \cdot \mathcal{K} - (|V|^2 - |V|)}. \quad (2.2)$$

$|V|$  bezeichnet die Ordnung<sup>9</sup> des Hypertextgraphs und nach Definition gilt  $\mathcal{C} \in [0, 1]$ . Es ist  $\mathcal{C} = 0 \iff \mathcal{H} = (V, \{\})$ . Weiterhin gilt  $\mathcal{C} = 1 \iff |E| = |V \times V| - |V|$ .  $(|V|^2 - |V|) \cdot \mathcal{K}$  ist der Maximalwert der Matricelemente aus der konvertierten Distanzmatrix. Er wird angenommen, falls  $E = \{\}$ .  $(|V|^2 - |V|)$  ist der minimale Wert der Summe der Matricelemente und wird erreicht, wenn  $\mathcal{H}$  der *vollständige Graph*<sup>10</sup> ist. Informell ausgedrückt bedeutet das, dass der Wert für das Gütemaß Compactness bezüglich einer bestimmten Hypertextstruktur Aufschluss darüber gibt, wie „dicht“ die Hypertextstruktur vernetzt ist. Ein hoher Compactness-Wert im Sinne von BOTAFOGO et al. sagt aus, dass von jedem Knoten aus jeder andere Knoten leicht erreicht werden kann. Als Beispiel betrachte man die Graphen aus Abbildung (2.2). Der erste Graph ist der vollständige gerichtete Graph  $K_4$  und

<sup>7</sup>Siehe Definition (4.1.5) in Kapitel (4.1).

<sup>8</sup>BOTAFOGO et al. setzen in ihren Untersuchungen  $\mathcal{K} = |V|$ .

<sup>9</sup>Die Ordnung eines Graphen ist die Anzahl der Knoten.

<sup>10</sup>Allgemein wird der vollständige Graph mit  $n$  Knoten in der Graphentheorie als  $K_n$  bezeichnet.

nach Gleichung (2.2) folgt  $\mathcal{C} = 1$ . Der zweite Graph besitzt die leere Kantenmenge, deshalb  $\mathcal{C} = 0$ . In [29] wurde von einigen Hypertexten der Compactness-Wert bestimmt und näher untersucht. So besaß beispielsweise die hypertextuelle Beschreibung des Fachbereichs Informatik der Universität Maryland CMSC (Computer Science Department at the University Maryland) einen Compactness-Wert von  $\mathcal{C}=0.53$ . Für das Buch in Hypertextform HHO (Hypertext Hands On!) [211] wurde der Wert  $\mathcal{C}=0.55$  ermittelt. Da es sich bei diesen Hypertexten um hierarchische, baumähnliche Graphen handelte, lag die Vermutung nahe, dass ein Compactness-Wert von ca. 0.5 typisch für solch strukturierte Hypertexte ist. Die Bildung eines Intervalls, in das man die Compactness-Werte von Hypertexten einordnen kann, um dann aus dem Wert innerhalb dieses Intervalls auf Gütemerkmale wie z.B. „gutes Navigationsverhalten“ zu schließen, ist jedoch aus Gründen der unterschiedlichen Interpretationsmöglichkeiten dieser Hypertextmetrik nicht möglich.

Für die Definition von Stratum betrachte man die Distanzmatrix von  $\mathcal{H}$

$$(\mathcal{D}_{ij})_{ij} := \begin{cases} w_{ij} & : \text{falls } w_{ij} \text{ existiert} \\ \infty & : \text{sonst} \end{cases}$$

$(\hat{\mathcal{D}}_{ij})_{ij}$  sei die Matrix, die man durch Ersetzung der Matrixelemente  $\infty$  durch 0 in  $(\mathcal{D}_{ij})_{ij}$  erhält. BOTAFOGO zeigt in [29], dass damit für Stratum  $\mathcal{S}$  die Gleichungen

$$\mathcal{S} = \begin{cases} \frac{4 \sum_{i=1}^{|V|} \left( \left| \sum_{j=1}^{|V|} \hat{\mathcal{D}}_{ji} - \sum_{j=1}^{|V|} \hat{\mathcal{D}}_{ij} \right| \right)}{|V|^3} & : \text{falls } |V| \text{ gerade} \\ \frac{4 \sum_{i=1}^{|V|} \left( \left| \sum_{j=1}^{|V|} \hat{\mathcal{D}}_{ji} - \sum_{j=1}^{|V|} \hat{\mathcal{D}}_{ij} \right| \right)}{|V|^3 - |V|} & : \text{falls } |V| \text{ ungerade,} \end{cases}$$

bestehen. Nach Definition von  $\mathcal{S}$  gilt  $\mathcal{S} \in [0, 1]$ .  $\mathcal{S} = 0$  bedeutet, dass die Hypertextstruktur in sich geschlossen und beispielsweise kreisförmig angeordnet ist.  $\mathcal{S} = 1$  beschreibt  $\mathcal{H}$  in Form einer vollständig linearen Graphstruktur. Wenn man zur gegebenen Hypertextstruktur die zugehörige Hierarchisierung betrachtet, drückt Stratum aus, wie tief und linear die hierarchische Struktur ist. Beide Maße, Compactness und Stratum, sind auf unmarkierten gerichteten Graphen definiert und beinhalten keinerlei semantische Relationen des vorgelegten Hypertextes. BOTAFOGO et al. führten diese Untersuchungen durch, indem sie von allen semantischen, pragmatischen und syntaktischen Typmerkmalen der hypertextuellen Träger abstrahierten. Ein bekanntes Phänomen von *quantitativen* Maßen zur strukturellen Charakterisierung von Hypertexten und zur Beschreibung von Hypertextnavigationsproblemen ist, dass die Ergebnisse solcher Maße oft vom konkret betrachteten Hypertext abhängen und mit anderen Messungen schlecht vergleichbar sind. Um diesem Problem entgegenzuwirken, führte HORNEY [114] eine weitere Untersuchung zur Messung von Hypertextlinearität, in Bezug auf

die Hypertextnavigation, durch. Dabei untersuchte HORNEY Pfadmuster, die durch bestimmte Aktionen der Benutzer im Hypertext erzeugt wurden, indem er Pfadlängen, ausgehend von den Knoten, bestimmte und mittelte. Dieses Prinzip wandte er auf das gesamte Hypertext-Dokument an und erhielt somit lineare Funktionen für diese Sachverhalte, die er als ein Maß für die Linearität eines Hypertextes definierte.

Neben BOTAFOGO et al. untersuchten und evaluierten auch DE BRA et al. [59, 60] Compactness und Stratum. Da in [29] Compactness und Stratum unter der Annahme definiert sind, dass im Hypertextgraph lediglich Vorwärtsbewegungen<sup>11</sup> ausgeführt werden, formulieren sie diese Maße neu, und zwar unter dem Aspekt, Backtracking-Bewegungen<sup>12</sup> im Hypertextgraph durchzuführen. Somit werden durch die modifizierten Metriken *navigational Compactness* und *navigational Stratum* von DE BRA et al. die Navigationseigenschaften von Benutzern in Hypertextstrukturen besser ausgedrückt.

Ebenfalls wurden die Auswirkungen von Compactness und Stratum auf das Navigationsverhalten in [151, 152] untersucht, indem aus den schon bekannten Maßen Pfadmetriken definiert und diese empirisch evaluiert wurden. Anstatt der in [29] definierten Matrizen verwendete MCENEANEY Pfadmatrizen für die analoge Anwendung dieser Hypertextmetriken. In der Pfadmatrix repräsentiert ein Matricelement die Häufigkeit von Knotenübergängen von einem Knoten zu jedem anderen Knoten im Navigationspfad. Diese Pfadmetriken ermöglichen aus graphentheoretischen Mustern, dargestellt durch Navigationspfade, die Navigationsstrategien von Hypertextbenutzern zu erkennen.

Eine Hypertextmetrik, welche Stratum ähnlich ist, wurde von COULSTON et al. in [55] definiert, indem sie die Navigationstiefe von Hypertextstrukturen basierend auf HUFFMAN-Codes vergleichen. Dabei stellt der HUFFMAN-Code einer Nachricht, dargestellt als Zeichenkette, die Binärcodierung jedes Zeichens der Nachricht dar, mit dem Ziel, dass die Länge der codierten Nachricht minimal ausfällt. Darauf basierend werden (i) die Informationen, die sich aus der Besuchsreihenfolge der Webseiten im Hypertextgraph ergeben, in einen HUFFMAN-Baum [118] transformiert, (ii) das codierte Navigationsverhalten des Benutzers wird in eine *Baumstruktur* transformiert, so dass diese mit dem erzeugten HUFFMAN-Baum strukturell vergleichbar ist. Um schließlich diese beiden Strukturen zu vergleichen, definieren COULSTON et al. ein Maß, welches das Benutzerverhalten mit einem optimalen Navigationsmuster, codiert durch den HUFFMAN-Code, vergleicht. Da-

---

<sup>11</sup> Im Sinne von BOTAFOGO et al. heißt das: Falls der Weg von  $v_i$  zu  $v_j$  nicht existiert, wird er mit der Konvertierungskonstante  $\mathcal{K}$  bewertet. Der Begriff des Weges wird in Definition (4.1.5) definiert.

<sup>12</sup>Das heißt, man folgt der gerichteten Kante  $(v_j, v_i)$ , falls man vorher die Bewegung  $(v_i, v_j)$  ausgeführt hat.



mit messen COULSTON et al. das Navigationsverhalten von Hypertextbenutzern gegen das durch den HUFFMAN-Code erzeugte Optimum.

Außer Compactness, Stratum und den bisher vorgestellten Maßen gibt es noch weitere graphentheoretische Maße im Hypertextumfeld. UNZ [235] beschreibt die zwei weiteren Maße *Density* und *Kohäsion*. Hauptsächlich gibt UNZ aber in [235] einen umfassenden Überblick über das Thema „Lernen mit Hypertext“, insbesondere bezogen auf Navigationsprobleme und die Informationssuche in Hypertexten. Density und Kohäsion wurden ursprünglich von WINNE et al. [247] eingeführt, um das Verhalten von Hypertextbenutzern im Zusammenwirken mit bestimmten Lernaktionen, wie z.B. „einen Text markieren“, „einen Text unterstreichen“ und „eine Notiz machen“ im Hypertextsystem STUDY graphentheoretisch zu analysieren. Um die spezifischen Graphmuster der Hypertextbenutzer zu gewinnen, bilden WINNE et al. formale Sequenzen von ausgeführten Lernaktionen in Adjazenzmatrizen<sup>13</sup> ab und erhalten so Graphmuster, die das Benutzerverhalten wiedergeben. Um dann messen zu können, welche Aktionen bei den Hypertextbenutzern welche Auswirkungen hatten, definierten WINNE et al. die Indizes

$$\mathcal{D} := \frac{\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} a_{ij}}{|V|^2}, \quad (\text{Density}) \quad (2.3)$$

und

$$\mathcal{COH} := \frac{\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} a_{ij} \cdot a_{ji}}{\frac{|V|^2 - |V|}{2}}. \quad (\text{Kohäsion}) \quad (2.4)$$

In den Gleichungen (2.3), (2.4) bezeichnet  $a_{ij}$  den Eintrag in der Adjazenzmatrix in der  $i$ -ten Zeile und der  $j$ -ten Spalte.  $\mathcal{D}$  gibt das Verhältnis der Anzahl der tatsächlich vorkommenden Kanten, zur Anzahl aller möglichen Kanten inklusive Schlingen [239] an und nach Definition gilt  $\mathcal{D} \in [0, 1]$ .  $\mathcal{COH}$  misst den Anteil von zweifach-gerichteten Kanten – das sind Kanten der Form  $(v_i, v_j), (v_j, v_i)$  für zwei Knoten  $v_i, v_j \in V$  – ohne Schlingen. Der Ausdruck  $\frac{|V|^2 - |V|}{2}$  gibt die Anzahl aller möglichen Knotenpaare an und es gilt ebenfalls  $\mathcal{COH} \in [0, 1]$ . Aus der Definition der Kohäsion schließen WINNE et al.: Je höher der Wert für die Kohäsion eines betrachteten Graphmusters ist, desto weniger schränken die Lernaktionen den Hypertextbenutzer ein. Genereller betrachtet kann man diese Maße als benutzer-spezifische Präferenzen innerhalb des Graphmusters interpretieren. Weitergehend und allgemeiner untersuchten NOLLER et al. [168] und RICHTER et al. [179] diese Problematik und entwickelten eine automatisierte Lösung zur Analyse von Navigationsverläufen. Die Navigationsmuster analysierten sie mit graphentheoretischen Mitteln und interpretierten sie ebenfalls als psychologische Merkmale wie z.B. gewisse Verarbeitungsstrategien, konditionales Vorwissen und benutzer-spezifische Präferenzen.

---

<sup>13</sup>Siehe Gleichung (4.1) in Kapitel (4.1.1).

Bis hierher wurden globale graphentheoretische Maße vorgestellt, die zur strukturellen Charakterisierung von Hypertext und zur Interpretation von Graphmustern dienen. Bekannt sind aber auch solche graphentheoretischen Maße, die zur Charakterisierung von Graphenelementen konstruiert wurden, insbesondere für die Knoten in einem Graph. Solche Maße sind in der Fachliteratur allgemeiner als *Zentralitätsmaße* bekannt und finden meist Anwendung in der *Theorie der sozialen Netzwerke* [204]. Sehr bekannte und grundlegende Arbeiten in diesem Bereich findet man bei HARARY [105] und HARARY et al. [106]. *Knotenzentralitätsmaße*, die etwas über die „Wichtigkeit“ und „Bedeutsamkeit“ von Knoten im Graph aussagen, wurden auch von BOTAFOGO et al. [29] definiert, bzw. bekannte Maße in einem neuen Kontext angewendet. So definierten sie die Maße

$$\text{ROC}_v := \frac{\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \mathcal{KDM}_{ij}}{\sum_{j=1}^{|V|} \mathcal{KDM}_{vj}}, \quad (\text{Relative Out Centrality})$$

$$\text{RIC}_v := \frac{\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} \mathcal{KDM}_{ij}}{\sum_{j=1}^{|V|} \mathcal{KDM}_{jv}}. \quad (\text{Relative In Centrality})$$

Dabei bedeuten  $\mathcal{KDM}_{ij}$  wieder die Einträge in der konvertierten Distanzmatrix, die durch die Definitionsgleichung (2.1) bereits angegeben wurde. BOTAFOGO et al. wandten das ROC-Maß an, um beispielsweise so genannte *Landmarks* – so werden identifizierbare Orientierungspunkte im Hypertext bezeichnet – zu kennzeichnen, weil Landmarks die Eigenschaft besitzen, mit mehr Knoten verbunden zu sein als andere Knoten im Hypertext. BOTAFOGO et al. kennzeichneten damit Knoten mit einem hohen ROC-Wert als Kandidaten für Landmarks. Dagegen sind Knoten mit niedrigem RIC-Wert im Hypertextgraph schwer zu erreichen. Letztlich dienen aber diese beiden Maße zur Analyse von Navigationsproblemen und damit wieder zum besseren Umgang mit dem „Lost in Hyperspace“-Problem.

Zum Abschluss dieser Übersicht wird eine Arbeit genannt, die ein graphentheoretisches Maß für den Vergleich von Hypertextgraphen liefert. So definierten WINNE et al. [247] das Maß *Multiplicity* für zwei gerichtete Graphen  $\mathcal{H}_1$  und  $\mathcal{H}_2$  als

$$\mathcal{M} := \frac{\sum_{i=1}^{|V|} \sum_{j=1}^{|V|} a_{ij} \cdot b_{ij}}{|V|^2} \quad i \neq j. \quad (2.5)$$

Nach Definition gilt  $\mathcal{M} \in [0, 1]$  und  $a_{ij}$  bzw.  $b_{ij}$  bezeichnen in Gleichung (2.5) die Einträge in der Adjazenzmatrix von  $\mathcal{H}_1$  bzw.  $\mathcal{H}_2$ . Dabei wird hier die Knotenmenge  $V$  als gemeinsame Knotenmenge der beiden Graphen angesehen und Multiplicity misst damit die Anzahl der gemeinsamen Kanten beider Graphen, relativ zur Anzahl aller möglichen Kanten. Die Motivation zur Definition von Multiplicity war, individuelle Taktiken und Strategien, die sich in zwei Graphmustern niederschlagen, vergleichbarer zu machen.

Eine Analyse von Hypertextstrukturen unter den Gesichtspunkten des Information Retrieval, in der auch Hypertextstrukturen anhand ihrer spezifischen Graphstruktur verglichen wurden, nahmen FURNER et al. [88] vor. Die graphentheoretischen Konstrukte, die in diesem Experiment angewendet wurden, waren z.B.

- Knotenindizes: Ein Beispiel ist der Ausgangs- und Eingangsgrad<sup>14</sup> eines Knotens.
- Graphindizes: Graphentheoretische Kenngröße für die strukturelle Beschreibung von Graphen. Beipielsweise wurde in [88] der aus der Chemie bekannte WIENER-Index [245] verwendet.

Das Hauptziel ihrer Untersuchung war jedoch die Aufdeckung von Zusammenhängen zwischen der Entstehung von Linkstrukturen und der Effektivität von *Hypertext Retrieval*-Systemen.

PARK stellt in [173] eine interessante Untersuchung der strukturellen Eigenschaften von Hypertextstrukturen vor, dessen Methoden von den bisher hier erwähnten abweichen. Er fasst Hypertextstrukturen als *formale Sprachen* auf und untersucht dann die von den Hypertextstrukturen erzeugten Sprachen und Grammatiktypen. PARK definiert dazu in [173] eine Grammatik  $G_1 = (V, \Sigma, P, \sigma)$ , wobei

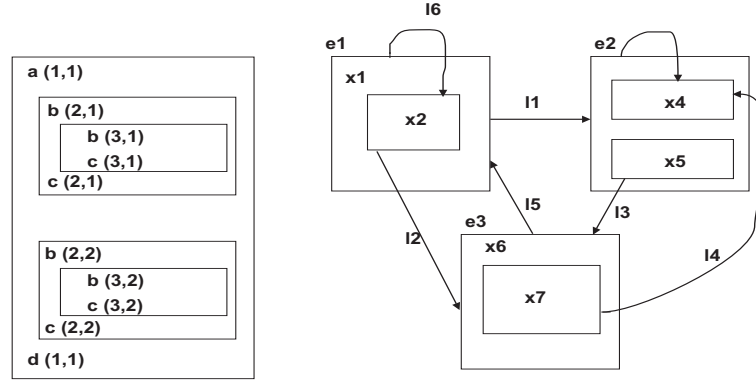
$$\begin{aligned} V &= \{\sigma, X, a, b, c, d\} \text{ (Alphabet),} \\ \Sigma &= \{a, b, c, d\}, \Sigma \subseteq V, \\ P &= \{\sigma \rightarrow aXd, X \rightarrow XbXcX, X \rightarrow \epsilon\}, \\ \epsilon &\text{ bezeichnet das leere Wort,} \\ \sigma &\text{ bezeichnet das Startsymbol.} \end{aligned}$$

Um den Aufbau einer Hypertextstruktur mit seiner Konstruktion zu erfassen, unterscheidet PARK zwischen der inneren Struktur – den *Hypertext Files* – und der äußeren Struktur eines Hypertextes. Hypertext Files können nun mit Wörtern modelliert werden, die von der Grammatik  $G_1$  erzeugt werden, also  $w \in L(G_1)$ . Die äußere Struktur, die aus einer Menge von Hypertext Files versehen mit einer Linkstruktur besteht, definiert PARK als  $HT = (E, X, L)$ . Dabei gilt:

- $E$  ist eine endliche Menge von Hypertext Files,
- $X$  ist die endliche Menge von allen *Matched Pairs* der Elemente von  $E$ ,
- $L$  ist die endliche Menge von geordneten Paaren von Matched Pairs in  $X$ .

---

<sup>14</sup>Siehe Definition (5.2.1) in Kapitel (5.2).



**Abbildung 2.3:** Das linke Bild zeigt das Hypertext File *abbccbbccd*. Jedem Symbol ist das Paar (Level, Ordnung) zugeordnet. Das rechte Bild zeigt einen Hypertext, der aus drei Hypertext Files *e1*, *e2* und *e3* besteht, zusammen mit seiner Linkstruktur. Beispielsweise enthält *e1* zwei Matched Pairs, nämlich *x1* und *x2*.

Das Konzept der Matched Pairs benötigt PARK, um verlinkbare Einheiten von Wörtern aus  $L(G_1)$  zu beschreiben. Um Matched Pairs in einem Wort zu identifizieren, wird in [173] das *Level* und die *Ordnung* von Symbolen in Wörtern  $w \in L(G_1)$  definiert. Das Level eines Symbols, das die Tiefe des Symbols im Wort angibt, kann ausgedrückt werden, indem die Produktionsmenge der Grammatik  $G_1$  in attributierter Form geschrieben wird. Die Abbildung (2.3) [173] zeigt schematisch ein Hypertext File zusammen mit einer Linkstruktur. Durch seine Untersuchung mit Beschreibungsmitteln aus der *Theorie der formalen Sprachen* erhält PARK schließlich neuartige Einblicke in strukturelle Aspekte von Hypertext, weil er ein nicht graphentheoretisches Beschreibungsmittel wählt und damit neue Modellierungsmöglichkeiten aufdeckt.

### 2.3.3 Zusammenfassende Bewertung

Die Abbildung (2.4) fasst die Ergebnisse des Kapitels (2.3.2) bewertend zusammen. Die Darstellungen in Kapitel (2.3.2) zeigen insgesamt, dass die Wirkung und die Aussagekraft von globalen Maßen zur strukturellen Charakterisierung von Hypertexten und zur Beschreibung von Graphmustern, z.B. Navigationsverläufe, beschränkt ist. Das liegt zum einen daran, dass einige der vorgestellten Maße für speziellere Problemstellungen entwickelt wurden oder in einer speziellen Studie entstanden sind, z.B. bei WINNE et al. [247]. Auf der anderen Seite erlauben quantitativ definierte Maße wie z.B. Compactness [29] keine allgemeingültigen Aussagen über eine verlässliche strukturelle Klassifikation von Hypertextgraphen bzw. über die Güte und Verwendbarkeit solcher Strukturen. Eine aussagekräftige Evaluierung der Maße und die Interpretation einer solchen Auswertung ist in vielen Fällen nicht erfolgt. Ein positiver Aspekt ist die durchgängig klare, ein-

Themenbereich	Literaturangaben	Positiv/Negativ
Indizes zur strukturellen HT-Charakterisierung (Compactness, Stratum)	[29, 55, 114]	Einfache Implementierbarkeit/Unzureichende Interpretierbarkeit
Indizes zur Beschreibung von HT-Lernaktionen (Density, Kohäsion)	[235, 247]	Einfache Implementierbarkeit/Unzureichende Strukturierung; Nur für Spezialfälle definiert
Struktureller Vergleich von HT-Graphmustern (Multiplicity)	[247]	Einfache Implementierbarkeit/Unzureichende Strukturierung
Knotenzentralitätsmaße für Hypertexte (ROC, RIC)	[29, 105]	Intuitive Definition/Lediglich auf unmarkierten Graphen definiert
Maße zur Beschreibung von HT-Navigationsverläufen	[29, 55, 59, 60, 114, 151, 152, 168, 179]	Einfache mathematische Modellierung/Unzureichende Interpretierbarkeit

**Abbildung 2.4:** Tabellarische Zusammenfassung der Ergebnisse aus Kapitel (2.3.2).

fache mathematische Modellierung und die leichte Implementierbarkeit, indem von komplexeren Typmerkmalen der Knoten und Links abstrahiert wird. Der negative Aspekt, der daraus unmittelbar resultiert, ist die fehlende semantische Information über solche Typmerkmale, die sich auch in der mangelnden Interpretierbarkeit von Werteintervallen innerhalb des ausgeschöpften Wertebereichs äußert.

### 2.3.4 Fazit

Für den Vergleich von Hypertextgraphen, im Hinblick auf lernpsychologische Implikationen, wurde das Maß Multiplicity von WINNE et al. [247], welches über der Kantenschnittmenge definiert ist, vorgestellt. Mit Multiplicity ist kein ganzheitlich struktureller Vergleich komplexer Hypertextgraphen möglich, da dieses Maß zu wenig von der gemeinsamen Graphstruktur erfasst. Wünschenswert wäre für den strukturellen Vergleich solcher Hypertextgraphen ein Modell, welches (i) möglichst viel von der gemeinsamen Graphstruktur erfasst und (ii) parameterisierbar ist, d.h. die Gewichtung spezifischer Grapheigenschaften ermöglicht. An dieser Stelle sei nun als Ausblick und Motivation für weitere Arbeiten die automatisierte Aufdeckung und die verstärkte Erforschung der graphentheoretischen

Struktur gerade für web-basierte Hypertexte genannt, weil (i) bisher wenig über deren charakteristische graphentheoretische Struktur und deren Verteilungen bekannt ist [193] und (ii) im Hinblick auf anwendungsorientierte Problemstellungen die Graphstruktur ganz besonders als Quelle zur Informationsgewinnung dienen kann. Das bedeutet, mit stetig wachsender Anzahl der hypertextuellen Dokumente im WWW werden Aufgaben wie die gezielte Informationsextraktion, das automatisierte *web-basierte Graphmatching* und die Gruppierung<sup>15</sup> ähnlicher Graphstrukturen für ein effizientes *Web Information Retrieval* [136] immer wichtiger. In Bezug auf das web-basierte Graphmatching wurde bereits das am Ende des Kapitels (2.2) skizzierte Verfahren erwähnt, welches in Kapitel (5) motiviert und entwickelt wird.

## 2.4 Existierende Clusteringverfahren zur Analyse hypertextueller Daten

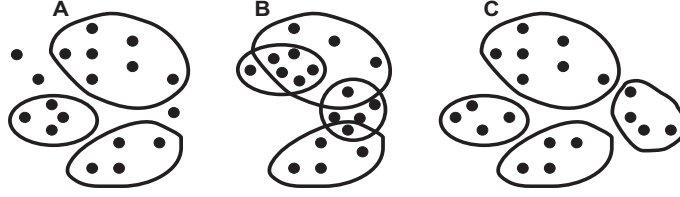
In Kapitel (2.3.2) wurden bekannte Arbeiten zur graphentheoretischen Analyse von Hypertextstrukturen vorgestellt. Dabei kamen auch Maße zur Beschreibung einzelner typischer Ausprägungen von Hypertexten und deren Anwendungen zur Sprache. Im Hinblick auf weiterführende graphentheoretische Methoden im Bereich des Web Structure Mining, wie das am Ende von Kapitel (2.2.2) skizzierte Verfahren, werden in diesem Kapitel eine Gruppe von multivariaten Analysemethoden, die Clusteringverfahren, vorgestellt. Bei den in Kapitel (2.3.2) dargestellten Verfahren stand die Charakterisierung typischer Ausprägungen graphbasierter Hypertexte auf der Basis numerischer Maßzahlen im Vordergrund. Im Gegensatz dazu gehören die Clusteringverfahren zur Gruppe der *Struktur entdeckenden* Verfahren, weil deren Ziel die Aufdeckung von strukturellen Zusammenhängen zwischen den betrachteten Objekten ist. Dabei ist die Einbeziehung mehrerer vorliegender Objektausprägungen die stark auszeichnende Eigenschaft von Clusteringverfahren [6]. Als weitere Anwendung innerhalb des Web Structure Mining und als eine Motivation für Kapitel (5.8) können Clusteringverfahren beispielsweise (i) zur Aufdeckung von *Typklassen* web-basierter Hypertexte eingesetzt werden, z.B. die Klasse der Mitarbeiterseiten innerhalb eines akademischen Webauftritts oder (ii) zur Trennung von strukturell signifikant unterschiedlichen Webseiten.

Clusteringverfahren [3, 6, 22, 23, 46, 78, 79, 122, 215, 218] werden zur Clustering von Objekten angewendet, um möglichst *homogene*<sup>16</sup> Cluster zu erzeugen.

---

<sup>15</sup>Die zu Grunde liegenden Verfahren der Datengruppierung heißen Clusteringverfahren. Siehe Kapitel (2.4).

<sup>16</sup>Die Clusterhomogenität wird in Kapitel (2.4.1) erklärt.



**Abbildung 2.5:** A: Disjunkte, aber nicht partitionierende Clusterung mit *nicht gruppierbaren* Objekten. B: Überlappende Clusterung. C: Partitionierende Clusterung

In der Regel ist bei Beginn der Clusterung die Anzahl der Cluster und die Clusterverteilung unbekannt, somit auch die Zuordnung der Objekte innerhalb der einzelnen Cluster. Clusteringverfahren sind deshalb im Bereich des *unüberwachten Lernens* [109] angesiedelt, weil sie ohne Lernregeln eine möglichst optimale Clusterung finden sollen. Die Clusterung erzeugt man, indem ähnliche Objekte in Clustern zusammengeschlossen werden mit dem Ziel, dass die Objekte der gefundenen Cluster eine ganz bestimmte *Charakteristik* aufweisen, bzw. jedes Cluster einen eigenen *Typ* repräsentiert. Abbildung (2.5) zeigt verschiedene Varianten von Clusterungen, die entweder je nach Anwendungsfall gewünscht sind oder deren Effekte, wie z.B. die Überlappung der Cluster, verfahrensbedingt auftreten.

Formeller ausgedrückt lässt sich diese Aufgabe für das Web Mining folgendermaßen beschreiben: Es sei  $D := \{d_1, d_2, \dots, d_n\}$ ,  $\mathbb{N} \ni n > 1$  die Menge der zu clusternden Dokumente. Will man die Clusteraufgabe in voller Allgemeinheit beschreiben, so fasst man die Dokumentenmenge als eine Menge  $O := \{O_1, O_2, \dots, O_n\}$  von unspezifizierten Objekten  $O_i$ ,  $1 \leq i \leq n$  auf. Eine Clusterung  $C_{fin}$  ist nun eine  $k$ -elementige *disjunkte Zerlegung* von  $D$ , also  $C_{fin} := \{C_i \subseteq D \mid 1 \leq i \leq k\}$ . Die Cluster  $C_i$  sollen dabei die Eigenschaft besitzen, dass, basierend auf einem problemspezifischen Ähnlichkeitsmaß  $s : D \times D \rightarrow [0, 1]$  (oder Abstandsmaß  $d : D \times D \rightarrow [0, 1]$ ), die Elemente  $d \in C_i$  eine hohe Ähnlichkeit zueinander besitzen, wohingegen die Elemente  $d, \tilde{d}$  mit  $d \in C_i \wedge \tilde{d} \in C_j, i \neq j$  eine geringe Ähnlichkeit zueinander besitzen sollen. Die Ähnlichkeits- oder Abstandsmaße basieren bei web-basierten Dokumentstrukturen zum einen direkt auf *inneren* (strukturellen) Eigenschaften eines Dokuments, wie z.B. die Darstellung gemäß des Vektorraummodells im Information Retrieval oder die graphbasierte Dokumentmodellierung. Zum anderen können auch *äußere* Merkmale, die ebenfalls über Ähnlichkeitsmaße zwischen den Dokumenten gemessen werden können, betrachtet werden, z.B. die *zip*-komprimierte Dokumentgröße [147]. Bei der Anwendung solcher Clusteringverfahren im Web Mining spielt die aus dem Information Retrieval bekannte *Cluster Hypothese* [46] eine wichtige Rolle. Sie sagt aus, dass die Ähnlichkeit zwischen relevanten und nichtrelevanten Dokumenten größer ist, als die Ähnlichkeit zwischen zufällig gewählten Teilmengen der Dokumentenmenge  $D$ . Verwendung findet sie bei der Konzeption und der Optimierung von Clusteringverfahren.

Clusteringverfahren besitzen außer dem Web Mining weiterhin vielfältige Anwendungsgebiete, z.B. die Datenanalyse in Unternehmen und die Mustererkennung in wissenschaftlichen Forschungsgebieten. In der Praxis des Web Mining finden oft *partitionierende* und *hierarchische* Clusteringverfahren Anwendung, wobei es noch eine Vielzahl anderer Verfahren gibt, z.B. *graphentheoretische*, *probabilistische* und *Fuzzy*-Clusteringverfahren [23, 78, 122]. Bevor ein Clusteringverfahren angewendet wird, ist es wichtig, die Ausprägungen der Beschreibungsmerkmale zu analysieren, um dann entscheiden zu können, ob zur Beschreibung der Unterschiede zwischen den Dokumenten ein Ähnlichkeits- oder ein Abstandsmaß gewählt wird. Die Frage nach der Lösung einer Clusteraufgabe stellt in der Regel ein Problem dar, da sie von der jeweiligen Anwendung und vom Verwendungszweck der Clusterung abhängt. Oft wählt man eine überschaubare Anzahl der gewonnenen Cluster aus, um sie entweder (i) aus der jeweiligen Anwendungsperspektive zu interpretieren oder (ii) sie mit statistischen Mitteln auf ihre Aussagekraft hin zu überprüfen. Generell sind die Anforderungen an moderne Clusteringverfahren hoch, da sie auf der Basis ihrer Konzeption möglichst viele Eigenschaften besitzen sollen, z.B.:

- Geringe Parameteranzahl.
- Einfache Interpretierbarkeit der Cluster.
- Gute Eigenschaften bei *hochdimensionalen* und *verrauschten* Daten.
- Die Verarbeitung von möglichst vielen Datentypen wie z.B. *ordinale* oder *nominale* Daten [128].

Jedoch ist nicht jedes Verfahren, das diese Eigenschaften besitzt, für eine Clusteraufgabe geeignet, weil die Verfahren gewisse Vor- und Nachteile besitzen, die in der Regel von den Daten, dem zu Grunde liegenden Ähnlichkeits- oder Abstandsmaß und der Konstruktion des Verfahrens abhängen. Dennoch wurden die meisten bekannten Clusteringverfahren theoretisch und praktisch intensiv untersucht, so dass sie gut voneinander abgrenzbar sind und somit die Auswahl eines Verfahrens für eine Clusteraufgabe leichter fällt.

### 2.4.1 Interpretation von Clusterlösungen

Um die Wirkungsweise von Clusteringverfahren besser zu verstehen, wird zunächst allgemein die Forderung der Cluster-*Homogenität*, die bereits in Kapitel (2.4) kurz erwähnt wurde, erläutert. Eine anschauliche Interpretation dieses Maßes, bezüglich eines Clusters  $C$ , liefert BOCK [23], indem er die Homogenität als numerische Größe  $h(C) \geq 0$  beschreibt, die angibt, wie ähnlich sich die Objekte in  $C$  sind oder anders formuliert, wie gut sich diese Objekte durch ihre



charakteristischen Eigenschaften beschreiben lassen. Ausgehend von einer Objektmenge  $O = \{O_1, O_2, \dots, O_n\}$ , einem Cluster  $C \subseteq O$  und einer Ähnlichkeitsmatrix  $(s_{ij})_{ij}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq n$ ,  $s_{ij} \in [0, 1]$ , gibt BOCK in [23] ein Maß für die Homogenität von  $C$  durch

$$h(C) := \frac{1}{|C| \cdot (|C| - 1)} \sum_{\mu \in I_C} \sum_{\nu \in I_C} s_{\mu\nu} \in [0, 1] \quad (2.6)$$

an, wobei  $I_C$  die entsprechende Indexmenge von  $C$  bezeichnet. Je größer  $h(C)$  ist, desto homogener ist  $C$  und umgekehrt. Ist anstatt der Ähnlichkeitsmatrix eine Distanzmatrix  $(d_{ij})_{ij}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq n$  gegeben, so sind

$$\begin{aligned} h_1^*(C) &:= \frac{1}{|C| \cdot (|C| - 1)} \sum_{\mu \in I_C} \sum_{\nu \in I_C} d_{\mu\nu}, \\ h_2^*(C) &:= \frac{1}{2|C|} \sum_{\mu \in I_C} \sum_{\nu \in I_C} d_{\mu\nu} \end{aligned}$$

Maße für die *Inhomogenität*, und es gilt hier: Je kleiner die Werte von  $h_i^*(C)$ ,  $i \in \{1, 2\}$  sind, desto homogener ist  $C$  und umgekehrt. BACKHAUS et al. [6] geben den  $F$ -Wert als Maß zur Homogenitätsbeschreibung eines Clusters  $C$  durch

$$F := \frac{\text{Var}(m_j, C)}{\text{Var}(m_j)}$$

an. Dabei beschreibt  $\text{Var}(x)$  die *Varianz* [6] einer Variablen  $x$ , wobei die Varianz als *statistisches Streuungsmaß* interpretiert werden kann. Der  $F$ -Wert ist das Verhältnis der Varianz der Merkmalsvariablen  $m_j$ , die eine Merkmalsausprägung des Objektes  $O_j$  ausdrückt, im zu prüfenden Cluster  $C$  zur Varianz von  $m_j$  in der Erhebungsgesamtheit. Kleine  $F$ -Werte drücken eine geringe Streuung der Variablen  $m_j$  im Cluster  $C$  aus, im Vergleich zur Streuung von  $m_j$  in der Erhebungsgesamtheit. In diesem Fall gilt das betrachtete Cluster  $C$  als homogen. Dagegen ist ein Cluster  $C$  als nicht homogen anzusehen, falls  $F > 1$  gilt, denn dies bedeutet umgekehrt, dass im Cluster  $C$  die Streuung von  $m_j$  höher ist als die von  $m_j$  in der Erhebungsgesamtheit.

In Anbetracht der großen Anzahl von existierenden Clusteringverfahren und unter Berücksichtigung ihrer Stärken und Schwächen, bezogen auf den jeweiligen Datenraum, ist eine Interpretation der gesamten Clusterlösung unbedingt notwendig. Allein die Qualität der Daten, eine mögliche Parametrisierung des zu Grunde liegenden Ähnlichkeitsmaßes, die Wahl des Clusterabstands und weitere Parameter des Clusteringverfahrens haben einen wesentlichen Einfluss auf die Clusterlösung und damit auch auf die Interpretation. Die Interpretation kann (i) mit Hilfe von numerischen Maßen zur Bewertung der Clusterhomogenität oder mit

Hilfe von gewählten Abbruchkriterien – denen meistens ebenfalls Homogenitätsbetrachtungen zu Grunde liegen – erfolgen oder (ii) durch Visualisierung und kreatives Interpretieren. Falls wieder eine Objektmenge  $O = \{O_1, O_2, \dots, O_n\}$  und die Ähnlichkeitsmatrix  $(s_{ij})_{ij}$  vorausgesetzt wird, so wäre beispielsweise die Maximierung der *mittleren Homogenität*  $h(C_{fin})$  [23] von  $C_{fin} := \{C_i \subseteq O \mid 1 \leq i \leq k\}$  eine mathematische Interpretation und somit auch ein Bewertungskriterium für die Güte einer Clusterlösung. Es ist

$$h(C_{fin}) := \max_{C_{fin}} \sum_{i=1}^k h(C_i),$$

wobei  $h(C_i)$  wieder durch die Gleichung (2.6) repräsentiert wird. Da je nach Anwendungsfall auch eine ausgepögte *Separation*, die Clustertrennung, gewünscht sein kann, ist die Maximierung des Ausdrucks [23]

$$A(C_{fin}) := \max_{C_{fin}} \sum_{i=1}^k \sum_{j=1}^k \alpha(C_i, C_j),$$

ein Maß für die Clustertrennung der Partition  $C_{fin}$ . Dabei bezeichnet  $\alpha(C_i, C_j)$  den Abstand<sup>17</sup> zwischen den Clustern  $C_i$  und  $C_j$ . Darüber hinaus gibt es weitere Möglichkeiten, um die Güte und die Aussagekraft von Clusterlösungen statistisch zu bewerten [6, 23, 122, 180]. Insgesamt gesehen kann oftmals das Ergebnis einer Clusterung als der erste Schritt betrachtet werden, um detailliertes Wissen über die betrachteten Objekte zu erlangen und um darüber hinaus eventuell neue Eigenschaften der Objekttypen zu erkennen. Weiterhin ist es notwendig, die Interpretation einer Clusterlösung vor einem speziellen Anwendungshintergrund zu sehen. Oder das Ergebnis der Clusterung stellt die Grundlage für eine weitergehende praktische Anwendung dar, da eine Clusterlösung, für sich isoliert betrachtet, keine weitreichende Aussagekraft besitzt.

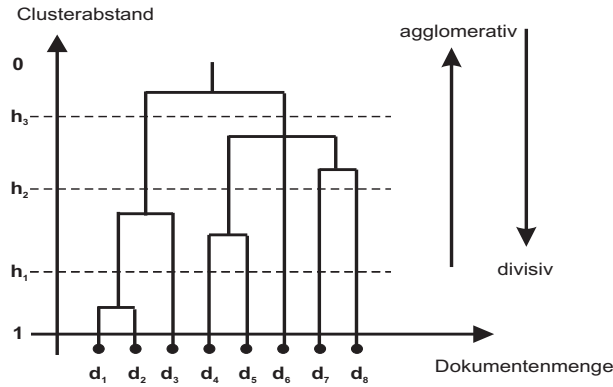
## 2.4.2 Hierarchische Clusteringverfahren

Um nun die grundlegende Funktionsweise von hierarchischen Clusteringverfahren für das Web Mining zu beschreiben, sei wieder die Dokumentenmenge  $D := \{d_1, d_2, \dots, d_n\}$  mit einem problemspezifischen Ähnlichkeitsmaß  $s : D \times D \rightarrow [0, 1]$  (oder Abstandsmaß) betrachtet. BOCK motiviert in [23] hierarchische Clusteringverfahren mit Eigenschaften der Homogenität in Bezug auf partitionierende Clusteringverfahren, bei denen  $C_{fin} := (C_1, C_2, \dots, C_k)$  die Eigenschaften einer *Partition*<sup>18</sup> von  $D$  erfüllt. Dabei ist offensichtlich, dass bei partitionierenden Verfahren (i) größere Homogenitätswerte der Cluster  $C_i$  durch eine größere

---

<sup>17</sup>Siehe Kapitel (2.4.2).

<sup>18</sup>Siehe Kapitel (2.4.3).



**Abbildung 2.6:** Dendrogramm für eine Clusteraufgabe mit acht Dokumenten. Die gestrichelten Linien deuten die gewählten Homogenitätsstufen an.

Kardinalität der Menge  $C_{fin}$  erreicht werden können, und umgekehrt (ii) sich hohe Homogenitätswerte nur bei hinreichend großer Kardinalität von  $C_{fin}$  erreichen lassen. Prinzipiell kann man zwei Arten von partitionierenden Verfahren unterscheiden: (i) Die Kardinalität der Menge  $C_{fin}$  ist vorgegeben oder (ii) die Homogenitätswerte der Cluster  $C_i$  werden von Anfang an durch Schranken gefordert. Dann ergibt sich im ersten Fall die Homogenität der Cluster durch das Verfahren selbst und im zweiten Fall ist  $k$  von der geforderten Ähnlichkeit innerhalb der Cluster abhängig. Da aber bei Clusteraufgaben die Zahl  $k$  und die Werte der Homogenitätsschranken in der Regel nicht bekannt sind, gelten beide der eben vorgestellten Möglichkeiten als nicht optimal. Hierarchische Clusteringverfahren versuchen dieses Problem dadurch zu lösen, dass sie eine Sequenz von Clusterungen erzeugen, mit dem Ziel, dass die Homogenitätswerte der Cluster mit wachsendem  $k$  steigen. Weiterhin gilt nach Konstruktion dieser Verfahren, dass immer homogenere Cluster dadurch gebildet werden, dass größere Cluster in kleinere unterteilt werden und dass dieses Prinzip beliebig nach unten fortgesetzt wird. Generell werden bei hierarchischen Clusteringverfahren *divisive* (top-down) oder *agglomerative* (bottom-up) Clusteringverfahren unterschieden, wobei sich in der Praxis die agglomerativen Verfahren durchsetzen. CHAKRABARTI [46] gibt eine Vorschrift in *Pseudocode* an, aus der die wesentlichen Konstruktionsschritte von agglomerativen Verfahren leicht zu erkennen sind:

1. Die initiale und damit die feinste Partion von  $D$  ist  $C_{fin} := \{C_1, C_2, \dots, C_n\}$ , wobei  $C_i = \{d_i\}$ .
2. **while**  $|C_{fin}| > 1$  **do**.
3. Wähle  $C_i, C_j \in C_{fin}$  und berechne den Abstand  $\alpha(C_i, C_j)$ .
4. Streiche  $C_i$  und  $C_j$  aus  $C_{fin}$ .

5. Setze  $\gamma := C_i \cup C_j$ .
6. Füge  $\gamma$  in  $C_{fin}$  ein.
7. **od while**

Das Ergebnis einer Clusterung mit hierarchischen Verfahren lässt sich als *Dendrogramm* visualisieren. Ein Dendrogramm einer fiktiven Clusterung zeigt die Abbildung (2.6). Dabei lassen sich nun auf jeder gewünschten Homogenitätsstufe  $h_i$  die Cluster ablesen und strukturell miteinander vergleichen. Man erkennt in Abbildung (2.6) deutlich ein auszeichnendes Merkmal eines agglomerativen Clusteringverfahrens: Auf der untersten Ebene stellen die Dokumente einelementige Cluster  $\{d_1\}, \{d_2\}, \dots, \{d_8\}$  dar; mit fallender Homogenität werden die Cluster auf den Ebenen immer größer, bis sie zu einem einzigen verschmolzen werden, welches alle Dokumente enthält. Ein weiteres wichtiges Merkmal eines hierarchischen Clusteringverfahrens liegt darin, dass Dokumente, die auf der Basis eines Ähnlichkeitsmaßes als sehr ähnlich gelten, sehr früh zu einem Cluster verschmolzen werden. Das ist aber gleichbedeutend damit, dass der dazugehörige Homogenitätswert  $h_i$  im Dendrogramm nahe bei Eins liegt. Weiterhin sind die Cluster auf den jeweiligen Homogenitätsstufen im Dendrogramm bezüglich ihrer inneren Struktur interpretierbar, da ein Cluster, das im Dendrogramm über mehrere Homogenitätsstufen in sich geschlossen bleibt, als sehr homogen angesehen werden kann. Wird dagegen ein Dokument erst im letzten oder vorletzten Schritt mit einem Cluster verschmolzen, so muss es auf Grund seiner Merkmale weniger ähnlich sein, als die Dokumente in einem sehr homogenen Cluster. Für das Ergebnis einer Clusteraufgabe, die mit einem hierarchischen Verfahren gelöst werden soll, ist aber auch die Güte der Daten, die Aussagekraft des zu Grunde liegenden Ähnlichkeits- oder Abstandsmaßes und vor allen Dingen die Wahl des Maßes  $\alpha$  entscheidend, um die Abstände  $\alpha(C_i, C_j)$  zweier Cluster zu berechnen. Ausgehend von einem Ähnlichkeitsmaß  $s : D \times D \rightarrow [0, 1]$  und den Clustern  $C_i$  und  $C_j$ , sind

$$\begin{aligned}\alpha_{SL}(C_i, C_j) &:= \min_{d, \tilde{d}} \left\{ s(d, \tilde{d}) \mid d \in C_i, \tilde{d} \in C_j \right\} \text{ (Single Linkage),} \\ \alpha_{AL}(C_i, C_j) &:= \frac{1}{|C_i||C_j|} \sum_{\tilde{d} \in C_i} \sum_{d \in C_j} s(d, \tilde{d}) \text{ (Average Linkage),} \\ \alpha_{CL}(C_i, C_j) &:= \max_{d, \tilde{d}} \left\{ s(d, \tilde{d}) \mid d \in C_i, \tilde{d} \in C_j \right\} \text{ (Complete Linkage)}\end{aligned}$$

gängige Clusterabstände.

Zusammenfassend formuliert ist die übersichtliche und anschauliche Darstellbarkeit des Ergebnisses in Form eines Dendogramms als positive Eigenschaft von

hierarchischen Clusteringverfahren zu sehen. Das Dendogramm, welches auch als Baumstruktur visualisiert werden kann, verlangt dabei nicht eine Clusteranzahl als Vorgabe, sondern auf jeder Ebene entsteht eine Anzahl von Clustern in natürlicher Weise. Weiterhin sind die einfache Implementierbarkeit und die gute Interpretierbarkeit der entstehenden Cluster als Vorteile von hierarchischen Verfahren zu werten. Für Daten, bei denen eine hierarchische Struktur zu erwarten ist, sind hierarchische Clusteringverfahren besonders sinnvoll. Da in der Regel dieses Kennzeichen nicht vorhanden ist, muss das Dendogramm für den jeweiligen Anwendungsfall interpretiert werden, da die hierarchische Struktur durch den Algorithmus erzwungen wird. Als Nachteil ist die Komplexität von hierarchischen Clusteringverfahren zu sehen, weil die Erzeugung der Ähnlichkeitsmatrix bereits quadratische Laufzeit besitzt und somit für Massendaten problematisch wird. Die Verwendung von verschiedenen Clusterabständen ist ebenfalls ein kritischer Aspekt, da Clusterabstände wie Single Linkage bzw. Complete Linkage oft die Tendenz zur Entartung haben, d.h. die Bildung von besonders großen bzw. kleinen Clustern.

### 2.4.3 Partitionierende Clusteringverfahren

In diesem Kapitel werden die Ziele und die grundlegende Wirkungsweise von partitionierenden Clusteringverfahren, die schon in Kapitel (2.4.2) kurz angesprochen wurden, erläutert. Wieder ausgehend von der Dokumentenmenge  $D$  und einem Ähnlichkeitsmaß  $s : D \times D \rightarrow [0, 1]$ , bildet die Menge  $C_{fin} := (C_1, C_2, \dots, C_k)$  eine partitionierende Clusterung von  $D$ , falls die Eigenschaften  $C_i \cap C_j, i \neq j$  (Disjunktheit) und  $\bigcup_{1 \leq i \leq k} C_i = D$  (volle Überdeckung der Menge  $D$ ) erfüllt sind. Basierend auf der vorgegebenen Menge  $D$  formuliert BOCK [23] die Hauptaufgabe der partitionierenden Clusteringverfahren als die Suche nach einer disjunkten, also nicht überlappenden Clusterung, welche die obigen Eigenschaften einer Partition besitzt und die auszeichnenden Merkmale der Dokumente optimal widerspiegelt. Weiterhin schlägt BOCK in [23] Ansätze zur Lösung dieses Problems vor, z.B.:

- Bereitstellung von statistischen oder entscheidungstheoretischen Modellen, mit denen die noch unbekannten Cluster und deren Objekteigenschaften als Parameter behandelt und abgeschätzt werden können.
- Einführung eines *Optimalitätskriteriums*, auf dem die *lokal optimale* Clusterung maßgeblich basiert.
- Initiale Festlegung von Startclustern und anschließende Konstruktion der gesuchten Cluster.
- Zuhilfenahme von daten- und anwendungsspezifischen Heuristiken.

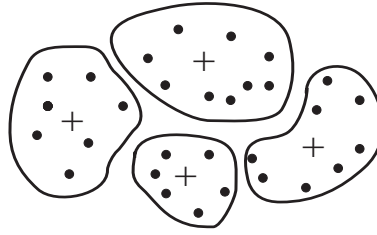
Bei partitionierenden Verfahren ist die finale Clusteranzahl  $k$  bei Beginn der Clustering nicht bekannt und die Dokumente  $d \in D$  werden ausgehend von gewählten Startclustern solange ausgetauscht, bis sich auf Grund eines Abbruchkriteriums eine möglichst lokal optimale Clusterung ergibt. Dagegen liegt bei der hierarchischen Clusterung auf jeder Hierarchiestufe eine eindeutige Menge von Clustern verfahrensbedingt vor, wobei diese Cluster nicht mehr aufgebrochen werden. Das in Theorie und Praxis bekannteste partitionierende Clusteringverfahren ist das *k-means*-Verfahren [18, 46, 215, 218], wobei es in verschiedenen Ausprägungen existiert, die sich meistens in der Art und Formulierung des Optimalitätskriteriums unterscheiden. Da *k-means* nur für quantitative Eingabedaten konzipiert ist, deren Abstände oft über die quadrierte *euklidische Distanz* berechnet werden, eignet sich für das *Dokumenten-Clustering* eine Abwandlung von *k-means*, das *k-medoids* Verfahren<sup>19</sup>. Anstatt von numerischen Startobjekten, die bei Beginn die Clusterzentren repräsentieren, wählt man in *k-medoids* Objekte (*Medoide*) aus  $D$  als Clusterzentren. Im weiteren Verlauf des Verfahrens werden lediglich die Ähnlichkeiten bzw. die Distanzen benötigt, um das Optimalitätskriterium in Form einer Zielfunktion und die neuen Medoids zu berechnen. Die wesentlichen Schritte von *k-medoids* lassen sich nachfolgend formulieren, wobei davon ausgegangen wird, dass die Dokumente  $d \in D$  in einer für das Clustering geeigneten Repräsentation vorliegen [104]:

1. Wähle zufällig  $k$  Dokumente als initiale Medoide und definiere damit die Menge  $M$  ( $|M| = k$ ).
2. **while** (no change) **do**.
3. Ordne jedes verbleibende Dokument dem nächsten Medoid zu (minimaler Abstand).
4. Wähle zufällig ein Dokument  $d_r \in D$ , das kein Medoid ist.
5. Berechne auf der Basis eines Kostenkriteriums  $c$  die Gesamtkosten  $S$  des Austauschs von  $d_r$  mit dem aktuellen Medoid  $d_{act}$ .
6. **if**  $c$  **then** tausche  $d_{act}$  mit  $d_r$  um eine neue Menge  $M$  von Medoiden zu bilden.
7. **od while**.

Abbildung (2.7) zeigt das fiktive Ergebnis eines partitionierenden Clusteringverfahrens. Vorteile von partitionierenden Clusteringverfahren wie *k-means* und *k-medoids* sind ihr intuitiver Aufbau und die einfache Implementierbarkeit. Als

---

<sup>19</sup>Die eigentliche Methode wird auch als PAM=Partitioning Around Medoids [224] bezeichnet.



**Abbildung 2.7:** Cluster mit ihren Zentren.

Lösungen liefern solche Verfahren aber nur *lokale Optima*, da mit einer anderen Startkombination eventuell eine bessere Clusterlösung berechnet werden könnte. Um diesem Problem entgegenzuwirken, bietet sich entweder eine Kombination mit anderen Clusteringverfahren oder eine iterierte Anwendung an. Ein Nachteil von beiden Verfahren,  $k$ -means und  $k$ -medoids, ist die Vorgabe der initialen Clusterzahl  $k$ , da diese in der Regel unbekannt ist. Eine weitere Schwäche von  $k$ -means ist die mangelnde *Robustheit* des Verfahrens, das heißt sein Verhalten bezüglich „Ausreißer“, da bei der Berechnung der quadrierten euklidischen Distanzen offensichtlich hohe Distanzwerte ermittelt werden und diese die Clusterbildung stark beeinflussen. Dagegen besitzt  $k$ -medoids eine schlechtere Komplexität in Bezug auf Massendaten, aber eine bessere Robustheit [109, 224].

#### 2.4.4 Sonstige Clusteringverfahren

Bisher wurden die hierarchischen und partitionierenden Clusteringverfahren detailliert vorgestellt, da diese Verfahren aus praktischen Gründen und auf Grund ihrer Interpretationsmöglichkeiten im Umfeld des Web Mining oft eingesetzt werden. In der Fachliteratur werden jedoch noch viele andere Clusteringverfahren behandelt, siehe z.B. [3, 23, 46, 78, 79, 122, 215, 218]. Darunter gibt es Verfahren und Modelle, die entweder in der Literatur oder in der Praxis ebenfalls einen hohen Bekanntheitsgrad erlangten. Einige werden im Folgenden skizziert:

- *Kombinatorische Clusteringverfahren:* Bei partitionierenden Verfahren ist die Frage nach einer Clustering  $C_{fin} = \{C_1, C_2, \dots, C_k\}$  der  $n$ -elementigen Dokumentenmenge  $D$ , wobei  $C_{fin}$  die Eigenschaften einer Partition besitzt. Die einfachste kombinatorische Lösung dieser Aufgabe wäre naiv dadurch zu bestimmen, dass man verschiedene Clusterlösungen auf der Basis eines Optimalitätskriteriums berechnet, z.B. ähnlich wie in  $k$ -medoids, in der Hoffnung, auf eine optimale Lösung zu stoßen. Betrachtet man jedoch die Anzahl der bei  $k$  Cluster möglichen Partitionen von  $D$ , so gilt [109, 218]:

$$S(n, k) = \frac{1}{k!} \sum_{j=0}^k (-1)^j \binom{k}{j} \cdot (k-j)^n$$

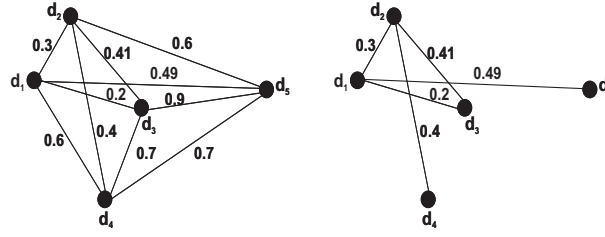
oder als Rekursionsgleichung

$$\begin{aligned}
S(n+1, k) &= S(n, k-1) + k \cdot S(n, k), \\
S(n, 1) &= 1, \\
S(n, n) &= 1, \\
S(n, 0) &= 0, \\
S(0, k) &= 0, \\
S(n, k) &= 0, \quad n < k.
\end{aligned}$$

In der *Kombinatorik* werden die  $S(n, k)$  als die STIRLING-Zahlen zweiter Art bezeichnet, beispielsweise ist  $S(10, 4) = 34.105$  und  $S(15, 4) = 42.355.950$ . Damit erkennt man, dass die Methode der *totalen Enumeration*, also das Ausprobieren aller möglichen Kombinationen, aus Komplexitätsgründen für praktische Zwecke unmöglich einsetzbar ist. Hinweise zur Optimierung kombinatorischer Verfahren und weitere Überblicke sind bei STEINHAUSEN et al. [218] und ARABIE et al. [4] zu finden.

- *Graphentheoretische Clusteringverfahren*: Ausgehend von der Dokumentenmenge  $D$  und einem problemspezifischen Abstandsmaß (ein Ähnlichkeitsmaß kann leicht in ein Abstandsmaß umgewandelt werden)  $d : D \times D \rightarrow [0, 1]$ , wird eine Abstandsmatrix  $(d_{ij})_{ij}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq n$  induziert, wobei  $d_{ij} \in [0, 1]$ . Diese Struktur kann, graphentheoretisch interpretiert, als ein kanten-markierter, vollständiger und ungerichteter Graph  $G_D = (V_D, E_D, f_{E_D}, A_{E_D}), f_{E_D} : E_D \rightarrow A_{E_D} := \{d_{ij} | 1 \leq i \leq n, 1 \leq j \leq n\}$  betrachtet werden. Nun interessiert man sich für *Umgebungen*, in denen auf Grund der Abstandswerte  $d_{ij}$  ähnliche Dokumente gruppiert werden und die Menge  $D$  somit auf diese Weise geclustert werden kann. BOCK [23] charakterisiert dieses Problem mit dem Begriff der  $d$ -Umgebung. Er versteht unter der  $d$ -Umgebung des Dokuments  $d_k \in D$  die Menge der Dokumente  $d_i \in D$ , deren Abstandswerte die Ungleichung  $d_{ik} \leq d$ ,  $d > 0$  erfüllen. Genauer formuliert definiert BOCK ein Cluster  $C \subseteq D$  als  $d$ -Cluster falls (i)  $C \neq \{\}$ , (ii)  $\forall d_k \in C$  gehört auch die  $d$ -Umgebung von  $d_k$  zum  $d$ -Cluster dazu und (iii) kein Cluster  $\tilde{C}$  mit  $\tilde{C} \subseteq C$  darf die Eigenschaften (i) und (ii) erfüllen. Man betrachte nun denjenigen Teilgraphen  $G_D^d = (V_D, E_D^d)$ ,  $E_D^d = E_D \setminus \{e = \{d_i, d_j\} | f_{E_D}(e) > d, \forall d_i, d_j \in V_D\}$  von  $G_D$ , für dessen Kantenmarkierungen die Ungleichungen  $f_{E_D}(e) \leq d, \forall e \in E_D^d$  gelten. BOCK beweist, dass die  $d$ -Cluster gerade die Zusammenhangskomponenten [108] des Teilgraphen  $G_D^d$  von  $G_D$  sind. Abbildung (2.8) zeigt beispielhaft für eine Menge  $D = \{d_1, d_2, \dots, d_5\}$  mit gegebener Distanzmatrix den vollständigen Graph  $G_D$  und den Teilgraph  $G_D^{0.5}$ . Ein wichtiges und einfaches graphentheoretisches Konstruktionsmittel für die  $d$ -Cluster ergibt sich sofort aus dem *minimalen Spannbaum* von  $G_D$ . Dabei ist der minimale Spannbaum gerade der Teilgraph  $B_D$  mit den Eigenschaften: (i)  $B_D$  ist ein Baum, (ii)  $B_D$





**Abbildung 2.8:**  $|D| = 5$ . Der vollständige Graph  $G_D$  und sein Teilgraph  $G_D^{0.5}$ .

enthält alle Knoten aus  $G_D$  und (iii) die Summe seiner Kantenmarkierungen fällt minimal aus. Die Konstruktionsmethode des minimalen Spannbaums und die anschließende Gewinnung der  $d$ -Cluster wird ausführlich in [23] beschrieben. Weitere graphentheoretische Clusteringverfahren werden in [79] vorgestellt. Je nach Anwendungsfall werden auch *Dichte-basierte* Clusteringverfahren verwendet, die auf Grund ihrer Konstruktionsweise sehr verwandt zu graphentheoretischen Verfahren sind. Sie werden in [79, 104] näher beschrieben. MEHLER [154] stellt einen Algorithmus zur *perspektivischen* Clustering ausgehend von so genannten *Kohäsionsbäumen* vor, die insbesondere der automatischen Textverlinkung dienen.

- *Probabilistische* Clusteringverfahren: CHAKRABARTI beschreibt in [46] Probleme des Clustering für web-basierte Dokumente in Bezug auf das Vektorraummodell. Algorithmen im Web Information Retrieval setzen voraus, dass die Elemente im Dokumentenraum zufälligen Prozessen unterliegen, wobei die Verteilungen innerhalb der Dokumente zunächst nicht bekannt sind. Probabilistische Clusteringverfahren ordnen die Objekte mit einer bestimmten Wahrscheinlichkeit einem Cluster zu, dabei ist aber in der Regel die Verteilung der Objekte und die Anzahl der Cluster unbekannt. Ein bekannter Algorithmus im Bereich der probabilistischen Clusteringverfahren ist der EM-Algorithmus (*Expectation Maximization*), der im Wesentlichen auf zwei Schritten beruht: (i) die Bestimmung der Clusterwahrscheinlichkeiten (Expectation) und (ii) die Parameterabschätzung der Verteilung mit dem Ziel, die Wahrscheinlichkeiten zu maximieren (Maximization). Der EM-Algorithmus wird, bezogen auf das Web Information Retrieval, ausführlich in [46] erklärt, wobei man weitere Überblicke in [18, 78, 79] findet.

## 2.5 Modellbildung: Polymorphie und funktionale Äquivalenz

In Kapitel (2.4) wurden bestehende Clusteringverfahren als Motivation für spätere Anwendungen im Web Structure Mining diskutiert. In dieser Arbeit wird

sich herausstellen, dass insbesondere die agglomerativen Clusteringverfahren ein wichtiges Bindeglied zur ähnlichkeitsbasierten Analyse web-basierter Dokumente darstellen.

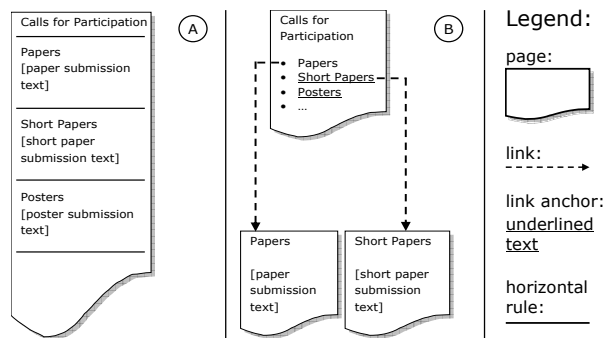
Im Hinblick auf die Erprobung eines neuen graphbasierten Repräsentationsmodells für web-basierte Dokumente, besitzt die bereits in der Einleitung formulierte These eine zentrale Bedeutung:

*Die graphbasierte Repräsentation hypertextueller Dokumente stellt einen zentralen Ausgangspunkt einerseits für graphbasierte Modellierungen und ähnlichkeitsbasierte Analysealgorithmen und andererseits für anwendungsorientierte Aufgaben im Web Structure Mining dar.*

Vorbereitend zur ähnlichkeitsbasierten Graphanalyse, werden zunächst Auswirkungen der inhaltsbasierten Kategorisierung im Rahmen des üblichen Vektorraummodells betrachtet. Auf der Basis einer grundlegenden Arbeit von MEHLER et al. [156], werden nun die Phänomene Polymorphie und funktionale Äquivalenz diskutiert. Darauf aufbauend thematisiert schließlich Kapitel (3) die aus der Polymorphie resultierenden Probleme während der inhaltsbasierten Kategorisierung web-basierter Dokumente. Die Ergebnisse des Kategorisierungsexperiments werden in Kapitel (3.5) dargestellt und interpretiert. Da das eigentliche Kategorisierungsexperiment im Rahmen des Vektorraummodells durchgeführt wurde, rechtfertigen die negativen Ergebnisse die Erprobung eines neuen graphbasierten Repräsentationsmodells. Um nun im Folgenden die Polymorphie und funktionale Äquivalenz zu erklären, wird zunächst der Begriff der *logischen Dokumentstruktur* kurz erläutert.

Der Begriff der „Dokumentstruktur“ wird in verschiedenen Kontexten gebraucht und umfasst z.B. die *physikalische Erscheinungsform* und die *logische Struktur* eines Dokuments. So sehen WILHELM et al. in [246] die messbaren Markierungen auf einem *Substrat*, z.B. Schrift auf Papier oder Zeichen auf dem Bildschirm, als die physikalische Erscheinungsform der externen Dokumentstruktur an. Dagegen beschreibt die logische Dokumentstruktur die von einem Autor gewollte Bedeutung der Dokumentbestandteile und fokussiert damit die inhaltliche Gliederung des Dokuments. Bezogen auf web-basierte Dokumente wird die logische Dokumentstruktur mit Auszeichnungssprachen wie z.B. XML oder HTML beschrieben. Somit besteht ein Dokument aus Elementen, die als die logischen Komponenten interpretiert werden, z.B. Autor, Titel und Überschrift.

MEHLER et al. [156, 157] stellten zur Analyse der *logischen Hypertext-Dokumentstruktur* ein Modell auf, welches in vier Ebenen gegliedert ist und auf dessen Grundlage die Unterscheidung von *Strukturtypen* und ihrer Instanzen beruht. Die Dokumenttypen repräsentieren dabei Websites, wie z.B. Wissenschaftler-



**Abbildung 2.9:** Schematische Darstellung zweier funktional äquivalenter Präsentationen [156]. Im Fall (A) auf der Basis einer Liste und (B) mit einem *compound document* [72], welches aus mehreren Webseiten besteht.

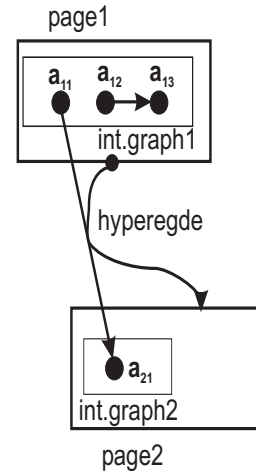
oder Konferenz/Workshop-Websites. In [156, 157] wurde dieses Vier-Ebenen-Modell anhand von englischsprachigen Konferenz-Websites veranschaulicht. Um problembezogene Aspekte der inhaltsbasierten Kategorisierung dieser Hypertexte zu untersuchen, wurde in [61, 156] ein Kategorisierungsexperiment durchgeführt. Bei den zu kategorisierenden Objekten handelte es sich um englischsprachige Webseiten von Konferenz/Workshop-Websites im Bereich Mathematik und Informatik. Analog der automatischen Textkategorisierung, deren Ziel in der Abbildung von Texteinheiten auf ein vordefiniertes Kategoriensystem besteht, liegt die Aufgabe der Hypertextkategorisierung in der Abbildung hypertextueller Einheiten auf ein statisch vorgegebenes Kategoriensystem. Im Mittelpunkt des Experiments stand nun die Untersuchung zweier Phänomene, die bei der inhaltsbasierten Kategorisierung web-basierter Hypertexte beobachtet wurden: Die Polymorphie und die funktionale Äquivalenz. Zur Erklärung der beiden Phänomene an einem Beispiel soll im Folgenden Abbildung (2.9) aus [156] betrachtet werden.

Die Abbildung zeigt zwei web-basierte Präsentationen, welche dieselbe Funktions- oder Inhaltskategorie „Calls for Participation“ manifestieren. Die linke Präsentation (A) manifestiert jedoch mehrere Funktions- oder Inhaltskategorien und ist somit nicht mehr eindeutig einer Kategorie zuzuordnen. Darauf bezieht sich die Polymorphie [61, 156]: Dasselbe Dokument besteht aus Ausdruckseinheiten, die mehrere Funktions- oder Inhaltskategorien manifestieren. Weiterhin wird hier die Funktions- oder Inhaltskategorie „Calls for Participation“ einmal mittels Präsentation (A) und im zweiten Fall, funktional äquivalent, auf der Basis der Präsentation (B) dargestellt. Präsentation (A) und Präsentation (B) gelten deshalb als funktional äquivalent, da verschiedene Bausteine web-basierter Komponenten ähnliche Funktionen realisieren können. Im Fall (A) wird die Dokumentenuntergliederung durch so genannte *horizontal rules* erreicht, im Fall (B) basiert die Aufgliederung auf der Basis von Links. In [156] wurde erstmalig die Hypothese formuliert, dass die Polymorphie und die funktionale Äquivalenz charakteristisch für web-basierte Hypertextstrukturen sind. Daraus ergeben sich aber unmittelbar

```

<gxl>
  <graph id="Testgraph" hypergraph="true">
    <node id="page1">
      <graph id="internalGraph1">
        <node id="anchor11"/>
        <node id="anchor12"/>
        <node id="anchor13"/>
        <edge from="anchor12" to="anchor13">
      </graph>
    </node>
    <node id="page2">
      <graph id="internalGraph2">
        <node id="anchor21"/>
      </graph>
    </node>
    <rel id="hyperedge">
      <relel target="page1" direction="in">
      <relel target="anchor11" direction="in">
      <relel target="page2" direction="out">
      <relel target="anchor21" direction="out">
    </rel>
  </graph>
</gxl>

```



**Abbildung 2.10:** (i): Das linke Bild zeigt einen fiktiven Hypergraph in GXL bestehend aus zwei Webseiten mit internen Graphen und einem Hyperlink. (ii): Das rechte Bild zeigt die graphentheoretische Struktur.

Probleme für die inhaltsbasierte Kategorisierung, die in Kapitel (3.5) interpretiert werden.

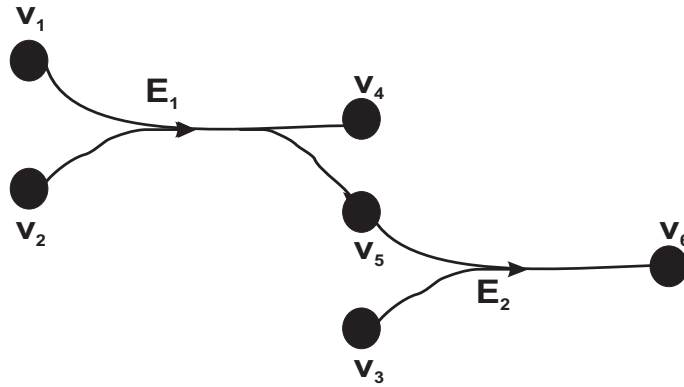
## 2.6 Konkreter Modellierungsansatz auf der Basis von GXL

In [156] wurde ein Modell für die Repräsentation von Hypertexten eingeführt, das zum einen funktional äquivalente Strukturtypen repräsentieren kann und zum anderen inhaltsorientierte Daten, z.B. Text und strukturelle Aspekte eines Hypertextes integriert. Basierend auf der hierarchischen (graphbasierten) Struktur einer Website führte das zu einem Modell, in dem (i) die hypertextuellen Einheiten durch Merkmalsvektoren repräsentiert werden und (ii) komplexe, interne und externe Linkstrukturen berücksichtigt werden. Dabei erfolgte die Realisierung des Modells in der XML-basierten Graphenaustauschsprache GXL (Graph exchange Language) [248]. Diese Beschreibungssprache für Graphen wurde mit dem Ziel entwickelt, ein standardisiertes Format für möglichst viele graphbasierte Anwendungen zu schaffen. Dieser Vorteil zeigt sich in der Vielfalt der Grapharten, die in GXL abbildbar sind: *typisierte, attributierte, gerichtete, geordnete, hierarchische Graphen und Hypergraphen*.

Um eine bessere Vorstellung über die Funktionsweise von GXL und deren Konstrukte zu gewinnen, werden einige Definitionsregeln für die Beschreibung der GXL-Graphtypen angeführt:

1. Das eigentliche Dokument ist mit den Tags `<gxl>` und `</gxl>` gekennzeichnet.
2. Jeder darin eingebettete Graph ist mit den Tags `<graph>` und `</graph>` gekennzeichnet. Darin muss das Attribut `"id"`, welches den Namen des Graphen spezifiziert, belegt sein. `"egdeids"`, `"egdemode"`, `"role"` und `"hypergraph"` sind weitere Attribute, die Standardwerte besitzen. Beispielsweise entscheidet das Attribut `"hypergraph"` ob der jeweilige Graph *Hyperkanten* enthält. Der Standardwert von `"hypergraph"` ist `"false"`. Über den Wert `"directed"/"undirected"` des Attributes `"egdemode"` kann eine Kante im Graphen als gerichtet oder ungerichtet definiert werden. Weitere Attribute in einem Graph werden auf der Basis von `"attr"`-Tags spezifiziert. Dabei wird der Attributname über `"name"` festgelegt und mögliche Wertetypen der Attribute sind z.B. `"string"`, `"float"` und `"bool"`. Container- und Mengentypen werden durch `"<seq>"` und `"<set>"` definiert, wobei diese Ausdrücke wieder ihre eigenen Typen enthalten können.
3. Hierarchien erzeugt man dadurch, dass innerhalb der Knoten und Kanten wieder Graphstrukturen eingeschachtelt werden. Hyperkanten werden in GXL mit den Tags `"<rel>"` und `"<relend>"` abgebildet. Mit `"<relend>"` werden die Endpunkte der Relation festgelegt. Der Wert des Attributes `"target"` markiert das zu verbindende Element und `"direction"` mit seinen Werten `"in"` und `"out"` legt die Richtung der Hyperkante fest.
4. Ein Graph kann mittels `<type>` auf ein im Dokument befindliches GXL-Schema [248] verweisen, welches dann als Schema-Graph im Dokument enthalten sein muss. Das Graph-Schema legt erst die eigentliche Bedeutung des Graphen in der Anwendung fest, indem es Regeln für z.B. Knoten- und Kantenbeziehungen und Attribute festlegt.

Als Beispiel zeigt die Abbildung (2.10) (i) das GXL-Codefragment eines fiktiven Hypergraphen mit deren eingeschachtelten Graphen und (ii) die graphentheoretische Struktur des Hypergraphen. In vielen bekannten Arbeiten werden Hypertexte nur als gerichtete und unmarkierte Graphen  $\mathcal{H} := (V, E)$ ,  $E \subseteq V \times V$  dargestellt [29, 235, 247]. Ein wesentlicher Nachteil dieses graphentheoretischen Modells ist, dass dabei die internen Linkstrukturen der Webseiten unberücksichtigt bleiben. Im Hinblick auf eine verstärkte Analyse web-basierter Hypertexte mit dem Hauptziel adäquatere Modellierungsmöglichkeiten aufzudecken, ist es jedoch notwendig Webseiten mit ihren internen Linkstrukturen abzubilden. In [156] wurde daher die GXL-Repräsentation der web-basierten Einheiten auf der Basis von attributierten, getypten, gerichteten und verschachtelten Hypergraphen realisiert. Die technische Umsetzung auf der Basis des HyGraph-Systems erfolgte in [94, 95].



**Abbildung 2.11:** Ein einfacher gerichteter Hypergraph mit den Hyperkanten  $E_1 = (\{v_1, v_2\}, \{v_4, v_5\})$ ,  $E_2 = (\{v_5, v_3\}, \{v_6\})$ .

Mathematisch stellen Hypergraphen eine Verallgemeinerung der gewöhnlichen Graphdefinition dar, da eine Hyperkante mehr als zwei Knoten verbinden kann. Dabei werden Knoten zu Kanten zusammengefasst, indem sie als Knotenmengen geschrieben werden. Während der Verallgemeinerung des Graphkonzepts gelangt man zunächst nur zu *ungerichteten Hypergraphen* [17], da durch die Zusammenfassung von Knoten zu Kanten die Richtungsinformationen verloren gehen. Ein ungerichteter Hypergraph  $\mathcal{H} = (V, E)$  ist auf einer endlichen Knotenmenge  $V := \{v_1, v_2, \dots, v_n\}$  definiert. Er besteht aus einer Menge  $E := (E_1, E_2, \dots, E_m)$ , wobei deren Elemente, die Hyperkanten, Teilmengen der Knotenmenge  $V$  mit den Eigenschaften (i)  $E_i \neq \{\}$ ,  $1 \leq i \leq m$  und (ii)  $V = \bigcup_{1 \leq i \leq m} E_i$  sind. Falls  $|E_i| = 2$ ,  $1 \leq i \leq m$ , so entartet  $\mathcal{H}$  zu einem gewöhnlichen Graph. Ein gerichteter Hypergraph [91] hingegen enthält nur gerichtete Hyperkanten in der Form von geordneten Paaren  $E = (X, Y)$ , wobei  $X$  als *tail* und  $Y$  als *head* der Hyperkante bezeichnet wird. Basierend auf dieser Definition können nun Attributmengen, Knoten- und Kantenalphabete mit ihren zugehörigen Markierungsfunktionen definiert werden, z.B. [231]. Abbildung (2.11) zeigt beispielhaft einen gerichteten Hypergraph.

## 2.7 Zusammenfassende Bewertung und Fazit

Ausgehend von einer kurzen Darstellung der Grundlagen von Hypertext und Hypermedia wurden in diesem Kapitel (2) im Wesentlichen Data Mining-Konzepte besprochen mit dem Ziel, sie auf bestehende und zukünftige Problemstellungen des Web Mining anzuwenden. Dabei wurden insbesondere bestehende graphentheoretische Methoden zur strukturellen Analyse von Hypertexten diskutiert, wobei deren zusammenfassende und kritische Bewertung bereits in Kapitel (2.3.2) erfolgte.

Clusteringverfahren	Literaturangaben	Positiv/Negativ
Hierarchische Clusteringverfahren	[23, 46, 78, 122]	Aussagekräftige und einfache Interpretierbarkeit/u.U. laufzeitintensiv
Partitionierende Clusteringverfahren	[23, 18, 46, 104, 224]	Intuitive Konstruktion; Einfache Implementierbarkeit/unbekannte Start-Clusteranzahl
Kombinatorische Clusteringverfahren	[4, 109, 218]	Intuitive Konstruktion; Einfache Implementierbarkeit/Oft unzureichendes Komplexitätsverhalten
Graphentheoretische Clusteringverfahren	[23]	Gute theoretische Fundierung/Kompliziert im praktischen Einsatz

**Abbildung 2.12:** Tabellarische Zusammenfassung der Clusteringverfahren aus Kapitel (2.4).

Ein weiterer Schwerpunkt von Kapitel (2) stellt Kapitel (2.4) dar. Das Hauptziel von Kapitel (2.4) bestand in der Diskussion und der kritischen Bewertung existierender Clusteringverfahren, um damit besser entscheiden zu können, welche Verfahren sich für zukünftige Problemstellungen im Web Mining eignen. Die Abbildung (2.12) fasst die dargestellten Clusteringverfahren und deren Eigenschaften bewertend zusammen. In dieser Arbeit werden schließlich auf Grund der guten visuellen Interpretierbarkeit von hierarchischen Clusteringverfahren, in Kombination mit mathematischen Hilfsmitteln zur Bewertung<sup>20</sup> von Clusterlösungen, speziell die agglomerativen Verfahren ausgewählt. Insgesamt betrachtet dienen sie für anwendungsorientierte Problemstellungen im Web Structure Mining, z.B. für die strukturorientierte Filterung web-basierter Dokumente, wobei sie bereits erzeugte Ähnlichkeitsmatrizen als Eingabe erhalten.

Betrachtet man allgemein die Anzahl der heute vorliegenden Clusteringverfahren, so erscheint die Auswahl eines geeigneten Verfahrens für den gewünschten Anwendungsfall schwer. Die Auswahl sollte sich auf jeden Fall an den vorliegenden Daten, am zu Grunde liegenden Ähnlichkeitsmaß und an der geplanten Weiterverwendung einer Clusterlösung orientieren. Zur Interpretation von Clusterlösungen wurden in Kapitel (2.4.1) mathematische Hilfsmittel vorgestellt. In Hinsicht auf die Clusterung strukturell ähnlicher Hypertexte ist es denkbar auch visuelle oder anwendungsbezogene Kriterien als zusätzliche Gütekennzeichen einer Clusterlösung zu definieren. Somit stellt eine Clusterlösung kein isoliert betrachtetes Ergebnis dar, sondern dient als Grundlage für Anwendungen im Web Structure Mining.

---

<sup>20</sup>Damit sind hier die Homogenitätsmaße aus Kapitel (2.4.1) und das Abbruchkriterium von RIEGER [180] gemeint.

In Vorbereitung auf die inhaltsbasierte Kategorisierung im Rahmen des bekannten Vektorraummodells, wurden in Kapitel (2.5) die Begriffe funktionale Äquivalenz und Polymorphie für web-basierte Einheiten definiert [156]. Der Begriff der funktionalen Äquivalenz bezieht sich dabei auf das Phänomen, dass dieselbe Funktions- oder Inhaltskategorie durch völlig verschiedene Bausteine web-basierter Dokumente manifestiert werden kann. Der Polymorphie-Begriff bezieht sich auf das Phänomen, dass dasselbe Dokument zugleich mehrere Funktions- oder Inhaltskategorien manifestieren kann. Die aus der Polymorphie resultierenden Probleme, während der inhaltsbasierten Kategorisierung web-basierter Einheiten, werden in Kapitel (3) ausführlich dargestellt.

Anstatt des bekannten Vektorraummodells stellt in dieser Arbeit das graphbasierte Repräsentationsmodell für web-basierte Dokumente den zentralen Ausgangspunkt für die ähnlichkeitsbasierte Graphanalyse dar. Das in Kapitel (2.6) vorgestellte Modell auf der Basis der Graphenaustauschsprache **GXL**, ist dabei die konkrete Realisierung der graphbasierten Repräsentation [156]. Die softwaretechnische Umsetzung innerhalb des **HyGraph**-Systems erfolgte in [94, 95]. Die positiven Aspekte des **GXL**-Repräsentationsmodell werden nun zusammenfassend formuliert:

- Repräsentation funktional äquivalenter Strukturtypen [156].
- Integration von inhaltsorientierten Daten (z.B. Text) und strukturellen Aspekten (hierarchische Graphstruktur).
- Hohe Flexibilität und große Vielfalt von modellierbaren **GXL**-Graphen.

Im Zuge der web-basierten Kommunikation wäre es für die zukünftige Entwicklung des Web Structure Mining besonders wünschenswert, neuere Methoden zur adäquaten Modellierung web-basierter Dokumente bereitzustellen. Im speziellen Umfeld des Web Structure Mining, in welchem mit graphentheoretischen Methoden Eigenschaften, Ausprägungen und sogar strukturelle Vergleiche hypertextueller Graphstrukturen bestimmt werden, besteht in der Zukunft besonderer Bedarf. Auf Grundlage des in Kapitel (2.6) vorgestellten **GXL**-Modells sind damit graphentheoretische Methoden angesprochen, mit denen eine aussagekräftige ähnlichkeitsbasierte Analyse möglich wird. Darauf basierend können aktuelle anwendungsorientierte Problemstellungen, z.B. die strukturorientierte Filterung und Fragen bezüglich zeitlich bedingter struktureller Veränderungen web-basierter Hypertextstrukturen, besser gelöst werden. Die Grundlage für die ähnlichkeitsbasierte Analyse im Bereich des Web Structure stellt in dieser Arbeit ein Graphähnlichkeitsmodell dar, welches in Kapitel (5) motiviert und entwickelt wird.



# Kapitel 3

## Grenzen der inhaltsbasierten Kategorisierung von Hypertextstrukturen

Obwohl sich die vorliegende Arbeit auf die struktur- und graphbasierte Analyse hypertextueller Dokumente konzentriert, wurden umfangreiche Teilarbeiten auch zur inhaltsbasierten Kategorisierung durchgeführt. Insbesondere erhält man dadurch ein besseres Verständnis für die Abgrenzung, sowie für die Möglichkeiten und Grenzen der beiden Teilbereiche. Im klassischen Sinne ist der wissenschaftliche Beitrag dieses Kapitels daher auch ein „Negativergebnis“. Wie im Verlauf des Kapitels klar wird, wurden dazu sowohl mathematisch-theoretische Arbeiten als auch softwaretechnische Entwicklungen und darauf aufbauende Experimente durchgeführt. Ausgehend von einer Motivation der Problemstellung in Kapitel (3.1) wird in Kapitel (3.2) die web-basierte Extraktion und die Konstruktion des verwendeten Testkorpus  $T_C$  detailliert dargestellt. Da das eigentliche Kategorisierungsexperiment auf der Basis eines maschinellen Lernverfahrens durchgeführt wurde, erfolgt dessen Motivation in Kapitel (3.3). In Kapitel (3.4) wird das Experiment mathematisch-theoretisch charakterisiert. Mit der Interpretation der Evaluierungsergebnisse und einem Fazit schließt dieses Kapitel ab.

### 3.1 Motivation

Das Ziel der Hypertextkategorisierung besteht darin, web-basierte Einheiten auf ein bestehendes System von Inhaltskategorien abzubilden [44]. In [156] wurde die Hypothese formuliert, dass Polymorphie und funktionale Äquivalenz charakteristisch für web-basierte Hypertexte sind. Im üblichen Rahmen der Hypertextkategorisierung wird die Zuordnung zwischen web-basierten Einheiten und Kategori-

en aber als funktional angesehen, das heißt, jede web-basierte Einheit, z.B. eine Webseite, wird höchstens einer Kategorie zugeordnet [44, 61, 156]. Unmittelbare Auswirkungen der Polymorphie wären Probleme bei der inhaltsbasierten Kategorisierung. Genauer formuliert würde sich dies darin äußern, dass Webseiten nicht mehr als eindeutig zu kategorisierende Einheiten anzusehen wären. Somit wäre die Kategorisierung im Sinne des üblichen Verständnisses nicht sinnvoll möglich. Um dies formal zu fassen, sei zunächst allgemein ein maschinelles Lernverfahren  $L$ , ein statisch definiertes Kategoriensystem  $K := \{K_1, K_2, \dots, K_{|K|}\}$  und eine Menge von Trainingsmengen  $T := \{T_1, T_2, \dots, T_{|K|}\}$  vorausgesetzt. Weiterhin bezeichnet  $U$  die endliche Menge der zu kategorisierenden Webseiten. Nach Anwendung des Lernverfahrens  $L$  wird die Zuordnung zwischen den Webseiten und den Kategorien nicht als Funktion, sondern in Form einer Relation  $\mathcal{R}$  erwartet. Sie kann wie folgt ausgedrückt werden [61]:

$$\mathcal{R} \subseteq U \times K, \mathcal{R} := \{(u, K_i) | U \ni u \text{ gehört zur Kategorie } K_i \in K\}. \quad (3.1)$$

Mit der experimentellen Aufdeckung dieser Relation beschäftigt sich das Kapitel (3.5).

Bekannte Untersuchungen im Bereich der Hypertextkategorisierung sind die Arbeiten von YANG et al. [252] und FÜRNKRANZ [87]. YANG et al. kategorisierten mehrere hypertextuelle Korpora unter verschiedenen Webseiten-Repräsentationen auf der Grundlage maschineller Lernverfahren, wie z.B. das Naive-BAYES [109] und das  $k$ -NN Verfahren [109]. Die Ergebnisse zeigten unter anderem, dass eine geeignete Repräsentation der hypertextuellen Daten entscheidend für den Erfolg einer solchen Studie ist. Weiterhin hatte sich die Einbeziehung von Metadaten positiv auf das Ergebnis der Kategorisierung ausgewirkt. Während zur inhaltsbasierten Kategorisierung üblicherweise die jeweiligen Informationen der zu kategorisierenden Einheit verwendet werden, beschreibt FÜRNKRANZ [87] einen Ansatz, der vom Erstgenannten abweicht: Textteile von Webseiten, die alle auf die Zielseite zeigen, werden extrahiert und zur Kategorisierung der Zielseite verwendet. Voraussetzung für diesen Ansatz ist, dass möglichst viele Webseiten als Referenzen auf die zu untersuchende Seite verweisen. Bei großen Datenmengen wären Klassifikationsaufgaben, beispielsweise im Bereich von Suchmaschinen, eine nützliche Anwendung des Verfahrens von FÜRNKRANZ.

Trotz dieser und weiterer Arbeiten ist die eigentliche Problemstellung der Hypertextkategorisierung nicht zufriedenstellend gelöst. Mögliche Gründe sind z.B.:

- Die gigantische Anzahl der im WWW existierenden Dokumente und deren inhaltliche und strukturelle Heterogenität.
- Die Tatsache, dass sich viele verschiedenartige Informationen, z.B. Plaintext und HTML-(Meta)Tags, aus den zu kategorisierenden Einheiten extrahieren lassen.

Dadurch bleiben die Hauptprobleme der Hypertextkategorisierung offen, die sich hauptsächlich in zwei schwierigen und noch unbeantworteten Fragestellungen widerspiegeln:

1. Wie müssen die in den Websites enthaltenen Informationen für die Kategorisierung angemessen repräsentiert sein?
2. Auf welche Weise sind Kategorisierungsverfahren zum optimalen Lernen von statistischen Mustern zur Hypertextkategorisierung einzusetzen?

Die erste Frage thematisiert eine geeignete Repräsentation der zu kategorisierenden Informationen. Dabei werden oftmals zur Hypertextkategorisierung die Methoden der Textklassifikation übertragen, indem Textklassifikationsverfahren auf der Basis einfacher Dokument-Repräsentationen, wie z.B. die Häufigkeitsverteilungen der vorkommenden Wörter angewendet werden. Man nimmt aber damit in Kauf, dass strukturelle Aspekte der zu kategorisierenden Einheiten vernachlässigt werden. Die zweite Frage zielt auf die Auswahl eines maschinellen Lernverfahrens, welches für die jeweilige Kategorisierungsaufgabe am optimalsten ist. Da jedoch diese Frage in der Praxis schwierig zu beantworten ist, sollte sich die Auswahl des Verfahrens auf jeden Fall an der speziellen Problemstellung und an der Art und Beschaffenheit der vorliegenden Daten orientieren.

Insgesamt will das Kapitel (3) auf der Grundlage der aufgestellten Hypothese des Kapitels - Polymorphie und funktionale Äquivalenz sind charakteristisch für web-basierte Hypertexte - die Grenzen der inhaltsbasierten Kategorisierung web-basierter Einheiten aufzeigen. Bestätigt sich diese These, so wären davon unmittelbar auch Bereiche des Web Mining, wie z.B. das Web Retrieval und die inhaltsbasierte Filterung, betroffen. Vor diesem Hintergrund müsste damit die verstärkte strukturelle Analyse und die adäquate Modellierung web-basierter Dokumente fokussiert werden, um negative Effekte bei der inhaltsorientierten Kategorisierung zu vermeiden.

Um im Folgenden die formulierte Hypothese mit Evaluierungsergebnissen zu untermauern, beschäftigt sich Kapitel (3.2) zunächst mit der Konstruktion des Testkorpus für das Kategorisierungsexperiment.

## **3.2 Das Testkorpus und die Extraktion web-basierter Hypertexte**

Das Experiment zielte auf die inhaltsbasierte Kategorisierung von englischsprachigen Konferenz/Workshop-Websites im Bereich Mathematik und Informatik

ab, wobei die Webseiten auf der Grundlage ihres textuellen Inhaltes<sup>1</sup> kategorisiert wurden. Die Schritte zur Erstellung des Testkorpus  $T_C$  lassen sich wie folgt darstellen:

1. Konstruktion der Linkmenge  $L_C$ : Auf der Basis eines im Rahmen dieser Arbeit entwickelten **Java**-Programms, welches ausgehend von einer Webseite alle darin befindlichen Links extrahierte, wurde mit Hilfe von englischsprachigen Konferenzkalender-Websites<sup>2</sup> eine entsprechende Menge von Konferenz-Links konstruiert. Es ist:  $|L_C| = 1000$ .
2. Extraktion der Hypertexte basierend auf  $L_C$ : Die Extraktion erfolgte auf Grundlage des im Rahmen der Arbeit entwickelten **HyGraph**-Systems [94, 95]. Unter Eingabe einer Start-Website und auf der Basis von Extraktionsfiltern, extrahiert **HyGraph** mit Hilfe eines implementierten *Webcrawlers* [96] alle beteiligten Webseiten und stellt die **GXL**-Repräsentation her. Für das *Parsing* der Webseiten wurde **HTMLParser** [257] aus der freien Software-datenbank **Sourceforge** verwendet. Um die vom Benutzer gewünschten Extraktionen auszuführen, war die Implementation von Extraktionsfiltern sinnvoll. Mögliche Filtereinstellungen sind [94, 95]:

- **StayOnSingleHost**: Nur Webseiten auf dem gleichen Host wie die Startseite werden vom Webcrawler in die Extraktion einbezogen.
- **StayOnSingleHostPath**: Nur Webseiten auf dem gleichen Host wie die Startseite und in einem Verzeichnispfad werden vom Webcrawler in die Extraktion einbezogen.
- **Unlimited**: Alle erreichbaren Webseiten werden vom Webcrawler in die Extraktion einbezogen.

Weiterhin ist auch eine *Mehrfachextraktion* möglich. In diesem Einstellungsmodus kann anstatt von nur einer Startseite eine Liste von Startseiten übergeben werden.

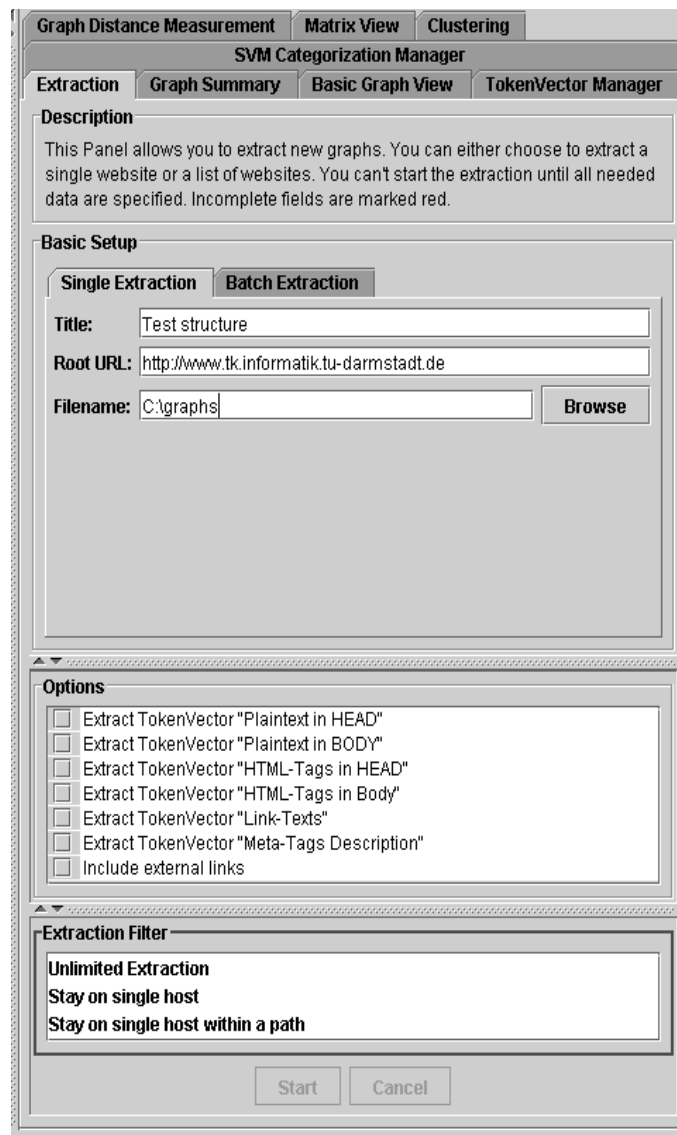
Zur Kategorisierung wurde als inhaltsbasierte Repräsentation der Webseiten das „*Bag of Words*“-Modell gewählt. Dabei handelt es sich um eine Kodierung der jeweiligen *Tokens*, wobei in diesem Fall die Häufigkeit der Tokens als Repräsentation ausgewählt wurde. Da aber bei Webseiten mehrere Tokenarten auftreten, wurden in **HyGraph** die folgenden Tokenarten unterschieden:

- **HTML**-Tags, die innerhalb von `<head>` eingeschlossen sind.
- **HTML**-Tags, die innerhalb von `<body>` eingeschlossen sind.

---

<sup>1</sup>Text-Tokens, die innerhalb von `<body>` eingeschlossen sind.

<sup>2</sup>z.B. <http://www.siam.org/meetings/calendar.htm>.



**Abbildung 3.1:** Konfigurationsbereich der Einzelextraktion von HyGraph. Im oberen Bereich werden die Grundeinstellungen der Extraktion festgelegt. Der mittlere Bereich der Maske legt die Tokenart und damit die jeweilige „Bag of Words“ fest. Mit Hilfe des unteren Konfigurationsbereichs wird der Extraktionsfilter gesetzt.

- Text-Tokens, die innerhalb von `<head>` eingeschlossen sind.
- Text-Tokens, die innerhalb von `<body>` eingeschlossen sind.
- Linktexte, die als Tokens repräsentiert werden.
- Tokens, die aus den Meta-Tags **Description** und **Keywords** stammen.

Abbildung (3.1) zeigt die Einstellungsmöglichkeiten von HyGraph bei einer Einzelextraktion.

3. Die Konstruktion der Linkstruktur: Der letzte Transformationsschritt, bevor die GXL-Repräsentation vollständig hergestellt ist, besteht aus der Berechnung der so genannten *Kernel-Hierarchie*, die von den *Kernel-Links* [94, 95] aufgespannt wird. Auf der Basis einer Heuristik werden, ausgehend von einem Startknoten, mit Hilfe einer *Breitensuche* die Kernel-Links bestimmt. Sie stellen das *Gerüst* der jeweiligen Website dar, welches graphentheoretisch einem *gerichteten Wurzelbaum* entspricht. Die Kernel-Hierarchie drückt die vom Hypertextautor beabsichtigte Navigationsstruktur der Website aus. Die gesamte Linkstruktur der betrachteten Graphen, die innerhalb von HyGraph berechnet wird, besteht aus den Kernel-Links, *Across-Links*, *Down-Links*, *Up-Links*, *External-Links* und *Reflexive-Links* (Schlingen). Die Veranschaulichung und die Formalisierung dieser Linktypen ist in Kapitel (5.2) zu finden.

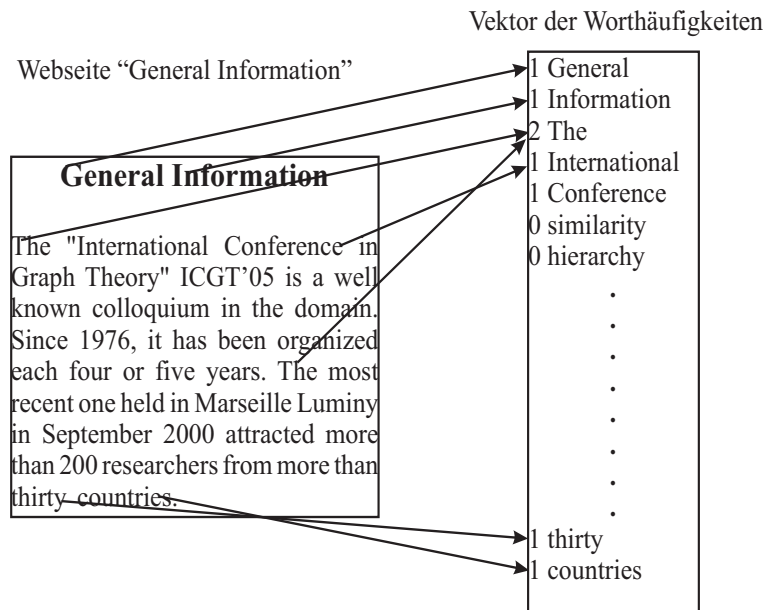
Ausgehend von der eingeführten Menge  $L_C$  ergab die Extraktion des Testkorpus  $T_C$ :  $|T_C| = 13481$ . Damit bestand das Testkorpus aus 13481 Webseiten.

### 3.3 Motivation des maschinellen Lernverfahrens

Im Bereich des *Text Mining* [155] werden Texte mit Hilfe von Data Mining-Verfahren analysiert. Um die Bedeutung der Texte zu reproduzieren, wird in der Regel das „Bag of Words“-Modell angewendet, in dem die Bedeutung als Häufigkeitsverteilung der vorkommenden Wörter (Tokens) aufgefasst wird. Abbildung (3.2) zeigt schematisch die Darstellung einer „Bag of Words“ an einer beispielhaften Webseite.

Die Kategorisierung der Webseiten erfolgte hier auf Grundlage einer „Bag of Words“ der Text-Tokens, die innerhalb des `<body>`-Tags eingeschlossen waren. Schwachpunkte, die sich beim praktischen Einsatz des „Bag of Words“-Modells ergeben, sind im Wesentlichen:

- Ordnungsbeziehungen zwischen den Tokens gehen verloren.



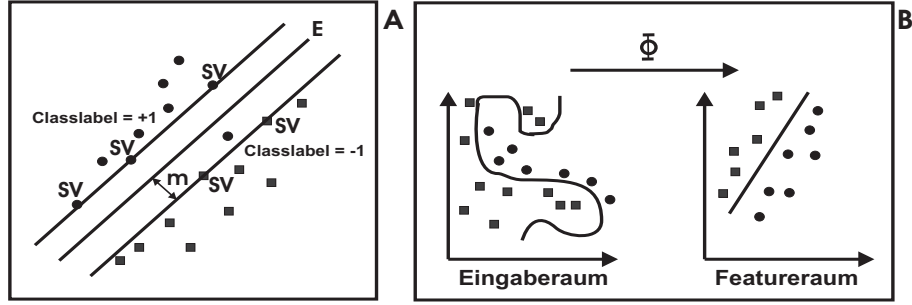
**Abbildung 3.2:** Vereinfachte schematische Darstellung der „Bag of Words“.

- Bei vielen Textklassifikationsproblemen entstehen *Feature-Vektoren* mit sehr hoher Dimension<sup>3</sup> (Dimensionsproblem).
- Benutzung unterschiedlicher Vokabularien und verrauschte Daten.

In Bezug auf das Dimensionsproblem besteht die Möglichkeit, eine *Dimensionsreduktion* [251] der Feature-Vektoren durchzuführen. Eine andere Möglichkeit, diesem Problem entgegenzuwirken, ist der Einsatz von *Kernel-Verfahren* [196], die auf Grund ihrer theoretischen Konzeption für hochdimensionale Kategorisierungsprobleme sehr geeignet sind. Vom maschinellen Lernverfahren unabhängig ist jedoch, dass der Kategorisierung eine gründliche Analyse des zu klassifizierenden Inhalts vorausgeht. Zum einen muss untersucht werden, ob der Inhalt überhaupt klassifizierbar ist. Das heißt, er muss sich möglichst eindeutig einer bestimmten Kategorie zuordnen lassen. Zum anderen muss das Kategoriensystem, welches für die Aussagekraft der Kategorisierungsaufgabe wesentlich ist, sinnvoll und repräsentativ gewählt werden.

Die *Support Vector Machine*-Kategorisierung (SVM) [56, 236], die für diese Kategorisierungsaufgabe verwendet wurde, ist ein bekanntes maschinelles Lernverfahren, das zur Gruppe der *Vektorraum-basierten Verfahren* gehört. Weiterhin gehört die SVM zur Klasse der *überwachten* maschinellen Lernverfahren und wurde insbesondere in der Textkategorisierung erfolgreich eingesetzt [125]. Dieses Lernverfahren erhält zunächst als *Eingabevektoren* bekannte Daten (Trainingsdaten),

<sup>3</sup>Oft ist die Vektorlänge  $> 50000$ .



**Abbildung 3.3:** A: Positiv (+1) und negativ (-1) klassifizierte Datenpunkte.  $m$  bezeichnet die maximale Trennschance. Die Supportvektoren, die die Lage der trennenden Hyperebene  $E$  maßgeblich bestimmen, sind mit SV markiert. B: Im linken Koordinatensystem ist keine lineare Trennung möglich. Schematische Transformation der Eingabedaten in den höherdimensionalen Featureraum mit anschließender linearer Trennung.

deren Klassenzugehörigkeiten bekannt sind. Darauf basierend wird mittels Entscheidungsregeln ein Modell *gelernt*, welches anschließend auf unbekannte Daten angewendet wird. Um das Grundprinzip der SVM kurz zu erklären, stellt man sich ein *linear separierbares Lernproblem* vor: Der Normalenvektor  $w$  und das Absolutglied  $b \in \mathbb{R}$  der trennenden Hyperebene  $E := \{x \in \mathbb{R}^n \mid \langle w, x \rangle + b = 0\}$  werden auf einer vorgegebenen Menge von Trainingsdaten  $T := \{(x_i, y_i) \mid x_i \in \mathbb{R}^n, y_i \in \{+1, -1\}, 1 \leq i \leq n\}$  so eingestellt, dass  $E$  die Trainingsdaten in zwei Klassen linear separiert. Gemäß einer Funktion  $\delta : \mathbb{R}^n \rightarrow \{\pm 1\}$  soll nun ein unbekannter Datenvektor korrekt klassifiziert werden. Gesucht ist jedoch die Hyperebene, die beide Datenklassen mit *maximaler Trennschance*<sup>4</sup> trennt, wobei diese Aufgabe als Optimierungsproblem formuliert werden kann [110]. Die Kernidee besteht darin, die Daten, die laut Annahme im Eingaberaum  $X$  in ihrer Ursprungsdimension nicht linear separierbar sind, auf der Basis einer Abbildung

$$\Phi : X \rightarrow F, \quad X \ni x \mapsto \Phi(x) = (\Phi_1(x), \Phi_2(x), \dots, \Phi_m(x)),$$

in einen höherdimensionalen *Featureraum*  $F$  zu transformieren. Anstatt im Eingaberaum  $X$  wird nun im Featureraum  $F$  eine lineare Trennung durch eine Hyperebene mit maximaler Trennschance  $m$  vorgenommen. Zur besseren Veranschaulichung wird die maximal trennende Hyperebene und die Transformation des Eingaberaums in den Featureraum in Abbildung (3.3) schematisch gezeigt. Unter der Voraussetzung, dass in  $F$  das Skalarprodukt<sup>5</sup>  $\langle \Phi(x), \Phi(y) \rangle$  definiert ist, ist das wesentliche Konstruktionsmittel der Hyperebene eine *Kernel-Funktion*  $k(x, y)$ . Ein Vektorraum  $X$  wird auch *innerer Produktraum* genannt, falls eine in beiden Argumenten lineare Abbildung  $\langle \cdot, \cdot \rangle : X \times X \rightarrow \mathbb{R}$  existiert, wobei für alle  $x, y \in X$  die Eigenschaften [56]

<sup>4</sup>Die maximale Trennschance wird auch als *Margin* bezeichnet.

<sup>5</sup>Allgemeiner auch *inneres Produkt* [83] genannt.



$$\langle x, y \rangle = \langle y, x \rangle, \quad (3.2)$$

$$\langle x, x \rangle \geq 0, \quad (3.3)$$

$$\langle x, x \rangle = 0 \iff x = 0, \quad (3.4)$$

gelten. Falls  $x, y \in \mathbb{R}^n$ , dann ist die Abbildung

$$\langle \cdot, \cdot \rangle: \mathbb{R}^n \times \mathbb{R}^n \longrightarrow \mathbb{R}, \langle x, y \rangle := \sum_{i=1}^n x_i \cdot y_i$$

ein inneres Produkt (Skalarprodukt) im  $\mathbb{R}^n$  mit den Eigenschaften (3.2), (3.3), (3.4). Weiterhin heißt

$$k(x, y) := \langle \Phi(x), \Phi(y) \rangle$$

Kernel-Funktion, falls für  $x, y \in X$  gemäß der Abbildung  $\Phi: X \longrightarrow F$  das Skalarprodukt  $\langle \Phi(x), \Phi(y) \rangle$  in  $F$  definiert ist. Man kann leicht zeigen, dass eine Kernel-Funktion die Eigenschaften eines Skalarproduktes besitzt, aber effizienter zu berechnen ist [56]. Dabei zeigt sich der wesentliche Vorteil der Kernel-Funktionen darin, dass die maximal trennende Hyperebene im Feature-Raum  $F$  ohne die explizite Anwendung der Transformationsfunktion  $\Phi$  bestimmt werden kann. Bekannte Kernel-Funktionen, die oftmals in Verbindung mit SVM's angewendet werden, sind für  $x, y \in X$

$$k(x, y) := \langle x, y \rangle \quad (\text{Linear}), \quad (3.5)$$

$$k(x, y) := (c \langle x, y \rangle + c_0)^d, \quad c, c_0 \in \mathbb{R}, d \in \mathbb{N} \quad (\text{Polynomial}), \quad (3.6)$$

$$k(x, y) := e^{-\gamma \|x-y\|^2}, \quad \gamma \in \mathbb{R} \quad (\text{Radial Basis Funktion}). \quad (3.7)$$

Ein Problem bleibt aber die Parameterauswahl in den Gleichungen (3.6), (3.7) bei einer konkreten Kategorisierungsaufgabe.

### 3.4 Charakterisierung des Kategorisierungsexperiments

In diesem Kapitel wird nun die spezielle Kategorisierungsaufgabe charakterisiert, indem die Schritte, angefangen mit der Konstruktion der Trainingsmengen bis zur optimalen Parameterbestimmung der Kernel-Funktion, detailliert erläutert werden. Es sei das Kategoriensystem

$$K := \{K_1, K_2, \dots, K_{|K|}\} \quad (3.8)$$

gegeben [61]. Die Funktions- oder Inhaltskategorien sind hier definiert<sup>6</sup> als  $K := \{\textit{submission and author instructions, call for papers, important dates, committees,}$

---

<sup>6</sup>Die Aufzählungsreihenfolge entspricht der Ordnung  $K_1, K_2, \dots, K_{13}$ .

*accepted papers, topics and general information, program, travel and accommodation, venue, invited speakers, registration, sponsors, workshops*},  $|K| = 13$ . Die Wahl dieser Kategorien wurde durch eine Linktextsuche bezogen auf das Testkorpus  $T_C$ <sup>7</sup> untermauert, indem jeder Linktext auf jeder Webseite von  $T_C$  genau einmal gezählt wurde. Ein hohes Vorkommen eines Linktextes im Hinblick auf die Kardinalität von  $T_C$  wurde dabei als Indikator für eine repräsentative Kategorie interpretiert.

Um nun das eigentliche Kategorisierungsexperiment vor dem Hintergrund des Dimensionsproblems, mit einer SVM durchzuführen, wurde aus praktischen Gründen die SVM-Bibliothek **LibSVM** [119] ausgewählt. Da die SVM-Klassifikation ursprünglich für rein binäre Probleme<sup>8</sup> entwickelt wurde und das gewählte Kategoriensystem (3.8) insgesamt 13 Kategorien enthält, kam in diesem Experiment die *Multiclass*-Strategie „*One Against All*“ zum Einsatz. Dabei wird das vorliegende 13-Kategorienproblem in 13 binäre Probleme unterteilt, indem für jede Kategorie ein SVM-Klassifikator gelernt wird. Dazu sind die *Klassen-Labels* der (positiven) Trainingsbeispiele für die entsprechende Kategorie auf +1 zu setzen, alle anderen (negativen) werden auf -1 gesetzt. Die Menge der Trainingsmengen  $T := \{T_1, T_2, \dots, T_{|K|}\}$ , die einen wesentlichen Einfluss auf das Kategorisierungsergebnis besitzt, wird im Folgenden konstruiert. Dabei wird die Konstruktion der Trainingsmenge  $T_i$  schrittweise erläutert:

1. Es seien  $\mathcal{L}_{T_i}(t) \in \{+1, -1\}, t \in T_i, 1 \leq i \leq |K|$  die Klassen-Labels der Beispiele  $t$  bezüglich der Trainingsmenge  $T_i$ . Weiterhin sei  $\mathcal{T}$  die Menge aller gelabelten Trainingsbeispiele und für deren Kategorie-Labels  $\mathcal{L}$  gilt:  $a \in \mathcal{T} : \iff \mathcal{L}(a) \in \{1, 2, \dots, |K|\}$ .
2. Definiere die Indexmenge  $I_{-i} := \{1, 2, \dots, i-1, i+1, \dots, |K|\}$  und es sei  $\mathcal{T}_{K_i} := \{a \in \mathcal{T} \mid \mathcal{L}(a) = i\}$ . Die Trainingsbeispiele in der Menge  $\mathcal{T}_{K_i}$  sind dabei die zukünftigen positiven (+1) Trainingsbeispiele für die gesuchte Menge  $T_i$ . Um die Mengen der Negativbeispiele (-1) zu konstruieren, definiert man weiterhin die Mengen  $\tilde{\mathcal{T}}_{K_j} := \{a \in \mathcal{T} \mid \mathcal{L}(a) = j\}, \forall j \in I_{-i}$ . Dabei setzen sich die Mengen  $\tilde{\mathcal{T}}_{K_j}$  wie folgt zusammen: Auf der Basis einer *einfachen Zufallsstichprobe* aus  $\mathcal{T}$  werden die zukünftigen Negativbeispiele gezogen, und es gelte die Bedingung

$$|\mathcal{T}_{K_i}| = \sum_{j \in I_{-i}} |\tilde{\mathcal{T}}_{K_j}|. \quad (3.9)$$

Aus der Gleichung (3.9) folgt, dass die zukünftigen Negativbeispiele der

---

<sup>7</sup>Siehe Kapitel (3.2).

<sup>8</sup>Ein bekanntes binäres Problem ist die Filterung von Spam-E-mails. Damit liegen genau zwei Klassen vor: Spam-E-mails (+1) und keine Spam-E-mails (-1).

Menge  $T_i$  auf die übrigen Kategorien mit annähernd gleicher Kardinalität verteilt wurden.

3. Damit ergibt sich direkt die Definition der Trainingsmenge  $T_i$ :

$$T_i := \{a \in \mathcal{T}_{K_i} \mid \mathcal{L}_{T_i}(a) = +1\} \bigcup_{j \in I_{-i}} \{a \in \tilde{\mathcal{T}}_{K_j} \mid \mathcal{L}_{T_i}(a) = -1\}. \quad (3.10)$$

Basierend auf Gleichung (3.10) wurde nun für jede Kategorie  $K_i, 1 \leq i \leq |K|$ , bezogen auf jede Trainingsmenge  $T_i$ , ein binärer SVM-Klassifizierer gelernt. Um auf der Grundlage der Trainingsmengen  $T_i, 1 \leq i \leq |K|$  Modelle  $m_1, m_2, \dots, m_{|K|}$  zur Vorhersage der noch unbekannten Webseiten  $u \in U$  zu erzeugen, müssen die optimalen Parametervektoren auf der Basis einer Kernel-Funktion bestimmt werden. Da für die konkrete SVM-Implementierung der SVM-Typ „C-SVM“ und ein RBF-Kernel in Form von Gleichung (3.7) verwendet wurde, müssen Parametervektoren der Form  $(C, \gamma)$  optimiert werden. Die Schritte der Parameteroptimierung sind wie folgt:

1. Definition des Suchraums:

$$P := \{(C, \gamma) \mid C = 2^g, \gamma = 2^s, g \in M_g := \{-4, 0, 4, \dots, 20\}, s \in M_s := \{-16, -12, -8, \dots, 8\}\}. \quad (3.11)$$

2. Basierend auf einer 5-fold „Cross Validation“ (CV) wurde für jede Trainingsmenge  $T_i$  der Klassifizierer mit allen Kombinationen der Parametermenge (3.11) aufgerufen mit dem Ziel, den CV-Fehler zu minimieren. Allgemein wird bei einer  $n$ -fold „Cross Validation“ die Trainingsmenge in  $n$  gleich große Teile (folds) aufgeteilt. Danach wird auf  $n - 1$  Teilmengen trainiert und auf der  $n$ -ten Teilmenge der *Generalisierungsfehler* [109] bestimmt. Führt man dieses Verfahren für alle möglichen  $n$  Unterteilungen durch und mittelt die jeweiligen CV-Fehler, so erhält man damit ein Kriterium für die Parameterauswahl.
3. Als Ergebnis der „Cross Validation“, bezüglich aller Parameterkombinationen erhält man direkt die Menge der quadratischen Matrizen

$$M_{CV} := \left\{ (\epsilon_{ij}^{T_1})_{ij}, (\epsilon_{ij}^{T_2})_{ij}, \dots, (\epsilon_{ij}^{T_{|K|}})_{ij} \mid 1 \leq i \leq |M_g|, 1 \leq j \leq |M_s| \right\},$$

wobei  $\epsilon_{ij}^{T_k}, 1 \leq k \leq |C|$  den CV-Fehler in der  $i$ -ten Zeile und der  $j$ -ten Spalte bezüglich  $T_k$  bezeichnet.

4. Um abschließend die Menge  $P_{fin}^{T_k}$  zu gewinnen, die den optimalen Parametervektor  $(C_{fin}, \gamma_{fin})$  für die Trainingsmenge  $T_k$  enthält, setzt man

$$P_{fin}^{T_k} := \left\{ (C_{fin}, \gamma_{fin}) \mid \epsilon_{fin}^{T_k} := \min_{\substack{1 \leq i \leq |M_g| \\ 1 \leq j \leq |M_s|}} \left\{ (\epsilon_{ij}^{T_k})_{ij} \right\} \right\}. \quad (3.12)$$

	Ist= +1	Ist= -1
Hyp( $u$ )=+1	$\delta_{11}$	$\delta_{12}$
Hyp( $u$ )=-1	$\delta_{21}$	$\delta_{22}$

**Abbildung 3.4:** Kontingenztafel für eine binäre Kategorisierung.

### 3.5 Interpretation der Evaluierungsergebnisse

Um zunächst die *Performance* der SVM-Kategorisierung zu evaluieren, wurden die Performancemaße Recall, Precision und *Accuracy* verwendet. Sie sind hier auf der Basis der Abbildung (3.4) definiert:

$$\begin{aligned} \text{Precision} &:= \frac{\delta_{11}}{\delta_{11} + \delta_{12}}, \\ \text{Recall} &:= \frac{\delta_{11}}{\delta_{11} + \delta_{21}}, \\ \text{Accuracy} &:= \frac{\delta_{11} + \delta_{22}}{\delta_{11} + \delta_{12} + \delta_{21} + \delta_{22}}. \end{aligned}$$

In der Abbildung (3.5) sind die Ergebnisse der Performancemessung dargestellt. Erwartungsgemäß zeigen die Ergebnisse sehr hohe Recallwerte und sehr niedrige Precisionwerte. Der Recallwert [176] kann als Wahrscheinlichkeit interpretiert werden, dass eine Webseite, die für eine Kategorie  $K_i$  relevant ist, vom Lernalgorithmus auch als solche gekennzeichnet wird. Mit Ausnahme der Kategorie  $K_7$  ( $K_7 \hat{=}$  *program*) weisen die hohen Recallwerte auf ein *konservatives Klassifikationsverhalten* hin. Das bedeutet, dass die Entscheidungen des Klassifizierers für eine Kategorie mit sehr hohem Recallwert immer richtig sind. Der Precisionwert [176] kann als die Wahrscheinlichkeit interpretiert werden, dass eine vom Lernalgorithmus als relevant gekennzeichnete Webseite wirklich relevant ist. Damit sagen die sehr niedrigen Precisionwerte in diesem Experiment aus, dass viele Webseiten vom Lernalgorithmus solchen Kategorien zugeordnet wurden, denen sie nicht angehören. Weiterhin ist der Accuracywert [125] als die Wahrscheinlichkeit anzusehen, dass eine Webseite, die zufällig aus der vorliegenden Verteilung der Beispiele gewählt wurde, vom Lernalgorithmus zur richtigen Kategorie zugeordnet wird. Auf Grund der Tatsache, dass die Webseiten-Repräsentation und somit die Kategorisierung auf dem bekannten Vektorraummodell basierte, folgt nun insgesamt aus Abbildung (3.5), dass die Kategorisierung sehr unsicher<sup>9</sup> und fehlerhaft ist. Auf Grund der niedrigen Precisionwerte liegt die Vermutung nahe, dass sich die ausgeprägte Falschkategorisierung in einer extremen Mehrfachzuordnung ausdrückt.

---

<sup>9</sup>Bezogen auf die hohen Recall- und niedrigen Precisionwerte.

$K_i$	Precision	Recall	Accuracy
$K_1$	29,1%	99,0%	70,8%
$K_2$	41,6%	99,0%	82,5%
$K_3$	41,2%	99,0%	90,4%
$K_4$	50,0%	99,2%	88,2%
$K_5$	66,6%	99,0%	72,1%
$K_6$	35,0%	99,1%	90,4%
$K_7$	25,5%	66,0%	68,4%
$K_8$	50,0%	99,2%	80,3%
$K_9$	32,0%	99,0%	66,3%
$K_{10}$	25,0%	99,0%	80,1%
$K_{11}$	46,1%	99,0%	71,3%
$K_{12}$	41,6%	99,0%	82,9%
$K_{13}$	52,1%	99,2%	94,1%

**Abbildung 3.5:** Ergebnisse der Performancemessung.

$K_i$	# matchings	$U_i$
$K_1$	2107	0,10
$K_2$	2661	0,05
$K_3$	1992	0,05
$K_4$	1546	0,24
$K_5$	3846	0,02
$K_6$	3616	0,02
$K_7$	2716	0,14
$K_8$	2245	0,03
$K_9$	3045	0,02
$K_{10}$	2206	0,01
$K_{11}$	3339	0,03
$K_{12}$	4627	0,03
$K_{13}$	1141	0,02

**Abbildung 3.6:** Ergebnisse der Auswertung von  $U_i$ .

Dies bestätigt sich durch die Evaluierung des Eindeutigkeitskoeffizienten [61, 156]  $U_i \in [0, 1]$ . Für die Definition von  $U_i$  gelte zunächst die Definition  $||K_i(u)|| = 1 :\iff u$  gehört zu Kategorie  $K_i$ . Der Koeffizient

$$U_i := \frac{|\{u \in U \mid K_i(u) \wedge \neg(K_1(u) \vee \dots \vee K_{i-1}(u) \vee K_{i+1}(u) \vee \dots \vee K_{|K|}(u))\}|}{|\{u \in U \mid K_i(u)\}|},$$

der die Eindeutigkeit der Kategorisierung von Webseiten  $u \in U$  bezüglich einer Kategorie  $K_i$  ausdrückt, ist informell definiert als das Verhältnis

$$\frac{\#\text{Webseiten } u \in U \text{ die ausschließlich zur Kategorie } K_i \text{ zugeordnet wurden}}{\#\text{Webseiten } u \in U \text{ die insgesamt zur Kategorie } K_i \text{ zugeordnet wurden}}.$$

Per Definition gilt: Je höher der Wert von  $U_i \in [0, 1]$ , desto eindeutiger ist die Kategorisierung der Webseiten bezogen auf die Kategorie  $K_i \in K$  und umgekehrt. Abbildung (3.6) zeigt die Ergebnisse der Auswertung von  $U_i$  basierend auf dem Kategoriensystem (3.8). Die sehr kleinen Werte von  $U_i$  bestätigen die extreme Mehrfachkategorisierung. Damit untermauert diese Auswertung die zentrale Hypothese nachhaltig, dass Polymorphie ein charakteristisches Phänomen für web-basierte Hypertexte, insbesondere für Webseiten des betrachteten Testkorpus  $T_C$ , ist. Folglich ist die inhaltsbasierte Hypertextkategorisierung des Testkorpus  $T_C$  nicht eindeutig. Dies bestätigt wiederum, dass die Zuordnung zwischen den Webseiten und den Kategorien zu einer Relation in Form von Gleichung (3.1), die in Kapitel (3.1) definiert wurde, entartet ist. Die Zusammenfassung der Interpretationsergebnisse implizieren nun insgesamt, dass das gewählte Vektorraummodell die komplexe Dokumentstruktur dieser Hypertexte nicht genügend erfasst und damit in diesem Zusammenhang unzureichend ist.

## 3.6 Fazit

### Praktische Aspekte des eingesetzten SVM-Lernverfahrens:

Bezogen auf die Evaluierung sind die Verarbeitung von Massendaten und ein gutes *Generalisierungsverhalten* als positive Aspekte für den Einsatz von SVM's zu nennen. Angewendet auf maschinelle Lernverfahren bedeutet letzteres in der statistischen Lerntheorie [109]: Extrahierte Gesetzmäßigkeiten aus vorliegenden Trainingsbeispielen sollen möglichst gut unbekannte Beispiele charakterisieren. Dagegen ist die langwierige Zusammenstellung der Trainingsmengen und die laufzeitintensive Parameterstudie negativ zu werten. Weiterhin ist in diesem Zusammenhang der hohe Zeitaufwand für die Datenvorverarbeitung zu nennen. Um das Laufzeitverhalten der Parameterstudie in dieser Untersuchung näher zu bestimmen, betrachte man die Optimierung der Parametervektoren  $(C, \gamma)$ . Daraus folgt aber zunächst allgemein, dass bei einer zweiparametrigen Optimierung mit den vorgegebenen Parametermengen  $M_1, M_2$  auf der Basis einer  $n$ -fold „Cross Validation“ ein  $|M_1| \cdot |M_2| \cdot n$ -maliges Aufrufen des SVM-Klassifikators pro Trainingsmenge  $T_i, 1 \leq i \leq |K|$  nötig ist. Mit den in Kapitel (3.4) definierten Mengen des Parameterraums (3.11) folgt damit, dass in dieser Untersuchung auf der Basis der 5-fold „Cross Validation“  $|M_g| \cdot |M_s| \cdot 5 = 7 \cdot 7 \cdot 5 = 245$  Aufrufe des SVM-Klassifikators, pro Trainingsmenge  $T_i, 1 \leq i \leq 13$ , erforderlich waren. Unter Berücksichtigung der hohen Dimension ( $\approx 50000$ ) der Feature-Vektoren war die Parameterstudie damit der laufzeitintensivste Teil der Evaluierung.

### Schlussfolgerungen aus dem Kategorisierungsexperiment:

Die Evaluierungsergebnisse der Abbildungen (3.5), (3.6) untermauern nachhaltig die These, die untersuchten Webpages des Testkorpus  $T_C$  seien systematisch durch Polymorphie gekennzeichnet. Die Webseiten aus  $T_C$  stammen dabei aus einem Bereich, der als besonders strukturiert angesehen wird. Dies kann deshalb angenommen werden, da die Funktions- und Inhaltskategorien wiederholt strukturierte Texteinheiten enthalten müssen. Die extrem geringe Trennschärfe zwischen den Kategorien  $K_i \in K$ , ausgedrückt durch Abbildung (3.6), könnte aber auch durch eine ungünstige Auswahl (i) der Features, (ii) des eingesetzten maschinellen Lernverfahrens oder (iii) des Kategoriensystems entstanden sein [61, 156]. Bezüglich (ii) wurde aber von JOACHIMS in [125] durch positive Evaluierungen der Performance gezeigt, dass die Textkategorisierung mit SVM's erfolgreich war. Daraus schließt JOACHIMS insbesondere, dass der SVM-Lernalgorithmus auf Grund seiner theoretischen Konzeption [236] die wesentlichen Merkmale einer Textkategorisierungsaufgabe, nämlich

1. hochdimensionale Featureräume,
2. die Existenz von wenigen nichtrelevanten Features und

### 3. dünn besetzte Feature-Vektoren,

geeignet berücksichtigt. Aus Punkt (2) der Aufzählung würde im Falle einer extremen Featurereduktion folgen, dass wichtige Klasseninformationen verloren gehen, und dass die Featurereduktion sogar eine Verschlechterung der Performance zur Folge hätte [125]. Diese Argumente belegen damit den sinnvollen Einsatz des SVM-Klassifikators für die vorliegende Untersuchung. Abschließend formuliert geben die negativen Aspekte des Kategorisierungsexperiments im Rahmen des Vektorraummodells Anlaß zur Erforschung eines neuen Repräsentationsmodells im Hinblick auf eine adäquatere Modellierung web-basierter Dokumente. Dabei handelt es sich in dieser Arbeit um das graphbasierte Repräsentationsmodell auf Basis der hierarchischen Graphstruktur, welches die Grundlage für ähnlichkeitsbasierte Analysen im Web Structure Mining bildet.

### Notwendige Schritte und Entwicklungen:

Für zukünftige Untersuchungen in diesem Problemkreis wäre es sinnvoll, weitere Kategorisierungsexperimente auf Grundlage neuer Testkorpora durchzuführen. Das Ziel solcher Untersuchungen wäre in erster Linie wieder der Nachweis der Polymorphie. Darüber hinaus sollten Experimente folgen, in denen die *Messbarkeit* der Polymorphie, also die numerische Bestimmung des Polymorphiegrades von Webseiten, im Vordergrund steht. Eine Beweisskizze für einen Polymorphiebeweis geben MEHLER et al. in [157]. Ein weiteres Experiment, in dem die Kategorisierung mit einem neuen Testkorpus von englischsprachigen Konferenz/Workshop-Websites durchgeführt wurde, ist bei MEHLER et al. [158] zu finden.

Auf der Interpretation basierend, dass das Vektorraummodell wegen mangelnder Strukturerrfassung komplexer Dokumentstrukturen zur inhaltsbasierten Kategorisierung ungeeignet ist, soll im Nachfolgenden als Alternative und im Hinblick auf das Kapitel (5) die Anwendung von Data Mining-Verfahren auf graphbasierte Repräsentationen web-basierter Hypertexte fokussiert werden. Auf der Basis einer geeigneten Graphrepräsentation<sup>10</sup> und eines aussagekräftigen Graphähnlichkeitsmodells [62, 73], welches in Kapitel (5) entwickelt wird, soll nun im Folgenden das web-basierte Graphmatching besonders thematisiert werden. Als Ergebnis der Berechnung der strukturellen Ähnlichkeit der Graphen von Webseiten-Testkorpora sollten auf Grundlage einer Graphmenge  $\bar{\mathcal{H}} := \{\mathcal{H}_1, \mathcal{H}_2, \dots, \mathcal{H}_n\}$  Ähnlichkeitsmatrizen der Form  $(s_{ij})_{ij}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq n$ ,  $s_{ij} \in [0, 1]$  resultieren. Sinnvolle Untersuchungen sind z.B.:

- Die Bestimmung der Verteilung der Ähnlichkeitswerte, wobei solche Verteilungen zur Klassifikation graphbasierter Hypertexte eingesetzt werden können.

---

<sup>10</sup>Diese wurde in Kapitel (2.6) bereits vorgestellt.

- Der Einsatz<sup>11</sup> von multivariaten Analyseverfahren, wobei speziell die Clusteringverfahren gewählt werden. Da die Clusteringverfahren Struktur entdeckende Verfahren sind, wird unter anderem die Auffindung von Clustern angestrebt, wobei die darin enthaltenen web-basierten Einheiten Instanzen eines eigenen strukturellen Objekttyps darstellen.

Im Hinblick darauf beschäftigen sich die nachfolgenden Kapitel mit der Motivation und der Entwicklung des Graphähnlichkeitmodells und dessen Anwendungen. Das Modell wird dabei praxisorientierte Aufgaben im Web Structure Mining übernehmen.

---

<sup>11</sup>Siehe Kapitel (5.8).



# Kapitel 4

## Graphentheorie und strukturelle Ähnlichkeit: Bekannte Methoden

Die Anwendung und die Entwicklung graphentheoretischer Methoden nehmen in dieser Arbeit einen großen Raum ein. Deshalb wird in diesem Kapitel in der gebotenen Kürze ein Überblick über die Graphentheorie und deren Anwendungsgebiete gegeben, um die in dieser Arbeit entwickelten Methoden fachlich einordnen zu können. Ausgehend von der Definition graphentheoretischer Begriffe, wird in diesem Kapitel weiterhin der Ähnlichkeits-Begriff hinsichtlich strukturierter Objekte erklärt. In Vorbereitung auf die Motivation und die Entwicklung des Graphähnlichkeitsmodells in Kapitel (5), erfolgt in Kapitel (4.1.2) eine ausführliche Diskussion bekannter Methoden zur Bestimmung der strukturellen Ähnlichkeit von Graphen. Kapitel (4.3) beschäftigt sich mit Graph Mining-Konzepten und bekannten Methoden zur Ähnlichkeitsbestimmung web-basierter Dokumentstrukturen.

### 4.1 Erforderliche Grundlagen

Der elementare Begriff des unmarkierten gerichteten Graphen  $\mathcal{H} := (V, E)$ ,  $E \subseteq V \times V$  wurde in der vorliegenden Arbeit bereits an einigen Stellen verwendet. In diesem Kapitel (4.1) werden darauf aufbauende graphentheoretische Begriffe definiert. Da die in Kapitel (5) entwickelten graphentheoretischen Methoden sich in erster Linie auf gerichtete Graphen beziehen, werden auch die folgenden graphentheoretischen Begriffe für endliche<sup>1</sup> gerichtete Graphen formuliert [120, 214].

---

<sup>1</sup>In dieser Arbeit werden nur endliche Graphen betrachtet, d.h.  $|V| < \infty$ .

**Definition 4.1.1 (Teilgraph)** Es sei  $\mathcal{H} := (V, E)$ ,  $E \subseteq V \times V$  ein (unmarkierter) gerichteter Graph gegeben. Dann heißt  $\mathcal{G} := (\hat{V}, \hat{E})$  mit  $\hat{V} \subseteq V$  und  $\hat{E} \subseteq E$  Teilgraph von  $\mathcal{H}$  und man schreibt  $\mathcal{H} \supseteq \mathcal{G}$ .

**Definition 4.1.2 (Induzierter Untergraph)** Es sei  $\mathcal{H} := (V, E)$  ein gerichteter Graph und  $\mathcal{G} := (\hat{V}, \hat{E})$  ein Teilgraph von  $\mathcal{H}$  gegeben. Gilt außerdem  $\hat{E} = E \cap (\hat{V} \times \hat{V})$ , dann heißt  $\mathcal{G}$  induzierter Untergraph von  $\mathcal{H}$ .

**Definition 4.1.3 (Markierter Graph)**  $\mathcal{H} := (V, E, f_V, f_E, A_V, A_E)$ ,  $E \subseteq V \times V$  heißt gerichteter und markierter Graph. Dabei bezeichnen  $A_V, A_E$  endliche nichtleere Knoten- und Kantenalphabete und  $f_V : V \rightarrow A_V$  und  $f_E : E \rightarrow A_E$  Knoten- und Kantenmarkierungsfunktionen.

Es ist klar, dass Teilgraphen und induzierte Untergraphen für markierte Graphen auf Basis der entsprechenden Einschränkungen von  $f_V$  und  $f_E$  ebenso definiert werden können.

Der Isomorphie-Begriff, der die strukturelle Äquivalenz von Graphen ausdrückt, besitzt in der Graphentheorie eine fundamentale Bedeutung. Weiterhin werden in Kapitel (4.2) bekannte Verfahren zur Messung der strukturellen Ähnlichkeit von Graphen angegeben, die auf Isomorphiebeziehungen beruhen. Daher wird der Isomorphie-Begriff nun formal definiert [214].

**Definition 4.1.4 (Graph-Isomorphie)** Es seien  $\mathcal{H} := (V, E, f_V, f_E, A_V, A_E)$  und  $\mathcal{G} := (\hat{V}, \hat{E}, \hat{f}_V, \hat{f}_E, A_V, A_E)$  markierte gerichtete Graphen.  $\mathcal{H}$  und  $\mathcal{G}$  heißen isomorph ( $\mathcal{H} \cong \mathcal{G}$ )  $:\iff$  Es existiert eine eindeutige Abbildung  $\phi$  von  $V \cup E$  auf  $\hat{V} \cup \hat{E}$  mit den Eigenschaften:

$$\begin{aligned} \phi(v) &\in \hat{V}, \quad \forall v \in V \\ \phi(e) &\in \hat{E}, \quad \forall e \in E \\ \phi((v, w)) &= (\phi(v), \phi(w)), \quad \forall v, w \in V, (v, w) \in E \\ f_V(v) &= \hat{f}_V(\phi(v)), \quad \forall v \in V \\ f_E(e) &= \hat{f}_E(\phi(e)), \quad \forall e \in E. \end{aligned}$$

Dabei heißt die Abbildung  $\phi$  Isomorphismus von  $\mathcal{H}$  auf  $\mathcal{G}$ . Informell erklärt sind zwei Graphen isomorph genau dann, wenn der eine aus dem anderen durch Umbenennung der Knoten hervorgeht.

**Definition 4.1.5 (Wege und Zusammenhang)** Es sei  $\mathcal{H} := (V, E)$ ,  $E \subseteq V \times V$  ein (unmarkierter) gerichteter Graph gegeben. Die Folge  $v_0, v_1, \dots, v_n$  heißt (gerichteter) Kantenzug, falls  $e_i = (v_i, v_{i+1}) \in E, i = 0, 1, \dots, n-1$ . Sind die

(gerichteten) Kanten  $e_i$  alle verschieden, so nennt man die Folge (gerichteten) Weg.  $v_0$  heißt Startknoten und  $v_n$  heißt Zielknoten. Im Fall  $v_0 = v_n$  heißt der Weg Zyklus, ansonsten handelt es sich um einen offenen (gerichteten) Weg.  $\mathcal{H}$  heißt zusammenhängend, wenn je zwei Knoten durch einen (gerichteten) Kantenzug verbindbar sind. Weiter heißt  $\mathcal{H}$  stark zusammenhängend, wenn für je zwei Knoten  $v$  und  $w$  immer ein (gerichteten) Kantenzug von  $v$  nach  $w$  existiert.

Abschließend für das Kapitel werden Graphen einer wichtige Graphklasse – die Bäume – definiert [186].

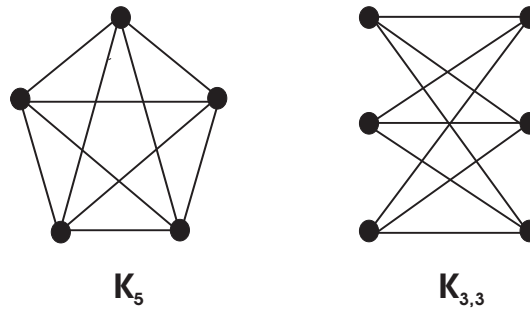
**Definition 4.1.6 (Baum)** Ein ungerichteter Graph heißt Baum, wenn er zusammenhängend und zyklensfrei ist. Ein gerichteter Graph heißt gerichteter Baum, wenn der zu Grunde liegende ungerichtete Graph ein Baum ist. Existiert darüber hinaus genau ein Knoten in den keine gerichtete Kante führt, dann wird der Graph Wurzelbaum genannt. Der auszeichnende Knoten heißt Wurzel.

Weitere graphentheoretische Begriffe werden speziell in den Kapiteln (5.2), (5.3) definiert. Sie werden im Wesentlichen zur Motivation und Modellierung des Graphähnlichkeitsmodells aus Kapitel (5) benötigt.

### 4.1.1 Überblick und Resultate der Graphentheorie

Der Graphenbegriff kommt in vielen wissenschaftlichen Bereichen, aber auch in normalen Lebensbereichen sehr häufig vor. Versucht man ein einfaches lokales Schienennetz aufzuzeichnen, entsteht ein Graph, indem man zum Beispiel „Punkte als Bahnhöfe“ und „Linien als Schienenstrecken“ andeutet. Der Graph des lokalen Schienennetzes kann auf einer nächsten Stufe der Formalisierung als eine Instanz des globalen Hauptproblems „Graphen aller Schienennetze“ aufgefasst werden. Anwendung findet die Graphentheorie heute in unzähligen Gebieten, z.B. der Informatik, der Elektrotechnik, der Soziologie, der Biologie und der Chemie. Es folgt nun ein kurzer Ausschnitt von Anwendungsfällen in den eben genannten Gebieten [89]:

- In der Informatik werden Graphen z.B. in den Bereichen Datenbankmodellierung, Netzwerktheorie und Hypermedia eingesetzt. In der Elektrotechnik finden sie z.B. Anwendung in der Darstellung von Platinenlayouts und Telekommunikationsnetzen.
- In der Soziologie werden graphentheoretische Konstruktionen in der Theorie der sozialen Netzwerke [105, 106, 204] und in der Stammbaum- und Ahnenforschung angewendet.



**Abbildung 4.1:** Der vollständige Graph  $K_5$  mit 5 Knoten und der bipartite Graph  $K_{3,3}$ .

- Die Chemie und die Biologie verwenden Methoden der Graphentheorie zur Darstellung und Bestimmung der Ähnlichkeit von Molekülstrukturen [10, 149, 212].

Die Ursprünge der Graphentheorie entstanden 1738 durch die Formulierung des bekannten Königsberger Brückenproblems durch EULER [77]. Dieses Problem löste er, indem er die eigentliche Problemstellung in die Sprache der Graphentheorie übersetzte. EULER [77] zeigte schließlich, dass kein Rundweg existiert, bei dem jede Brücke genau einmal überquert wird.

Die *moderne* Graphentheorie ist stark durch das Teilgebiet der *topologischen* Graphentheorie [237] geprägt, in der die strukturelle Untersuchung eines Graphen im Vordergrund steht. Die Ursprünge der topologischen Graphentheorie gehen auf EULER zurück, der 1750 die Polyederformel entdeckte. Erst durch KURATOWSKI fand diese Formel 180 Jahre später Anwendung. Dabei benutzte er sie zur Charakterisierung von *planaren* Graphen. Ein Graph ist genau dann planar, wenn er so gezeichnet werden kann, dass sich die Kanten des Graphen nicht schneiden. Die Kanten dürfen sich lediglich in den Knoten des Graphen berühren. Als Ergebnis seiner Untersuchungen bewies er den bedeutenden

**Satz 4.1.7 (Satz von Kuratowski)** *Ein endlicher Graph ist genau dann planar, wenn er keine Unterteilung von  $K_5$  oder  $K_{3,3}$  als Teilgraphen enthält. Allgemein heißt ein Graph  $H$  Unterteilung eines Graphen  $G$ , wenn er aus  $G$  durch das sukzessive Einfügen von endlich vielen neuen Knoten gewonnen werden kann.*

Dabei stützt sich die Aussage von Satz (4.1.7) [120] auf die nicht planaren Graphen  $K_5$  und  $K_{3,3}$ , die in Abbildung (4.1) dargestellt sind. Weiter wurden in [237] Zusammenhänge zwischen der *reinen Topologie* [123] und der Graphentheorie untersucht, indem VEBLEN Graphen als *simpliziale Komplexe* [202] auffasst und Graphen mit Hilfe dieser Strukturen klassifiziert.

Ein weiterer Forschungsgegenstand, der ebenfalls der topologischen Graphentheorie angehört, sind *Graphminoren*. Ein Graph ist *Minor* eines Graphen  $G$ , wenn

er durch Entfernen von Kanten und Knoten sowie durch das Zusammenziehen von Kanten aus  $G$  entsteht [181]. Insbesondere interessiert sich die Graphentheorie dabei für Strukturaussagen und -klassifikationen von Graphen auf der Basis von Minoren. Als bekannte Anwendung lässt sich mit Hilfe der Definition des Graphminors eine äquivalente Formulierung von Satz (4.1.7) angeben.

**Satz 4.1.8** *Ein Graph  $G$  ist genau dann planar, wenn weder  $K_5$  und  $K_{3,3}$  Minoren von  $G$  sind.*

Ein ebenfalls gut erforschtes Teilgebiet der Graphentheorie ist die *algebraische Graphentheorie* [98]. Ihr Hauptproblem besteht darin, Graphen mit Hilfe von algebraischen Strukturen, z.B. Matrizen und Gruppen, darzustellen, um die daraus resultierenden Eigenschaften algebraisch auszudrücken. Viele Forschungsarbeiten in der algebraischen Graphentheorie widmen sich der Untersuchung von Polynomen auf Graphen, z.B. chromatisches Polynom, TUTTE-Polynom und Rangpolynom [8, 98]. Sie bilden so genannte Invarianten, deren Auffindung für die Unterscheidung von Graphen sehr wichtig ist. Dabei heißt eine Funktion auf Graphen Graph-Invariante, falls die Funktion isomorphen Graphen gleiche Werte zuordnet. Ein ebenfalls wichtiges Gebiet dieser Theorie ist die Erforschung von *Zufallsgraphen* [25], wobei die ersten bedeutenden Arbeiten von ERDÖS und RÉNYI [75, 76] stammen. Um nach ihren Überlegungen Zufallsgraphen zu konstruieren, starteten sie mit einer beliebigen Knotenmenge  $V$ ,  $|V| = n$  und erzeugten für jede Kombination von zwei Knoten Kanten mit der Wahrscheinlichkeit  $p \in [0, 1]$ . Als Erwartungswert für die Kardinalität der Kantenmenge  $E$  folgt damit  $\frac{n(n-1)}{2}$ . Ein breites Anwendungsfeld für Zufallsgraphen bieten die Untersuchungen [65, 141, 142] des WWW-Graphen, wobei man mit solchen Modellen besonders das Wachstum beschreiben möchte.

Die Untersuchung des *Spektrums* eines Graphen, welche ebenfalls der algebraischen Graphentheorie zuzuordnen ist, soll als letztes Forschungsgebiet der modernen Graphentheorie erwähnt werden. Um den Begriff des Spektrums kurz zu erklären, betrachte man zu einem beliebigen Graphen  $G = (V, E)$  die Adjazenzmatrix

$$\mathcal{A} := \begin{cases} 1 & : (v_i, v_j) \in E \\ 0 & : \text{sonst} \end{cases} \quad (4.1)$$

Das Spektrum von  $G$  besteht nun aus den beiden Mengen  $M_\lambda = \{\lambda_1, \lambda_2, \dots, \lambda_k\}$  und  $M_n = \{n_1, n_2, \dots, n_k\}$ . Dabei bezeichnet  $M_\lambda$  die Menge der Eigenwerte der Matrix  $\mathcal{A}$ , und  $M_n$  ist die Menge der Vielfachheiten<sup>2</sup>. Oft wird das Spektrum auch als Nullstellenmenge des *charakteristischen Polynoms*  $\chi(\lambda) = \det(\mathcal{A} - \lambda E)$  formuliert. Ein großer Teil der Untersuchungen in der *spektralen Graphentheorie* hat

---

<sup>2</sup>Bezogen auf die Nullstellenmenge  $M_\lambda$ .  $n_i$  bezeichnet die Vielfachheit der Nullstelle  $\lambda_i$  von  $\chi(\lambda)$ .

zum Ziel, Zusammenhänge zwischen dem Spektrum und der Struktur eines Graphen aufzudecken. Diese Fragestellung wurde ausführlich in [58] untersucht und hat viele bekannte Ergebnisse hervorgebracht. Graphspektren besitzen ebenfalls interessante Anwendungen in der Informatik, z.B. spektrale Clusteralgorithmen in der Bildverarbeitung [244]. Einen guten Überblick über die spektrale Graphentheorie erhält man in [52, 58].

Als wichtige Mitbegründer der modernen Graphentheorie sind neben EULER [77], KURATOWSKI [143], ERDÖS [75, 76] auch HALIN [103] und SACHS [186, 187] zu nennen. Eine Einführung in die Grundlagen der Graphentheorie ist z.B. in [14, 26, 66, 108, 194, 229, 232] zu finden.

### 4.1.2 Ähnlichkeit strukturierter Objekte

Will man die „Ähnlichkeit“ zwischen bestimmten Objekten feststellen, so muss zunächst die Auswirkung dieses Begriffs diskutiert werden, weil nicht unmittelbar klar ist, was man unter dem Begriff „Ähnlichkeit“ verstehen soll. Sollen beispielsweise unterschiedliche Testpersonen die Frage

*Ist Objekt  $O_1$  ähnlich zu Objekt  $O_2$ ?*

beantworten, so besitzen verschiedene Testpersonen eine wahrscheinlich unterschiedliche Vorstellung von der Ähnlichkeit zwischen diesen beiden Objekten. In der Regel bezieht sich der Begriff der Ähnlichkeit auf unterschiedliche Aspekte. Somit unterscheidet SOBIK [213] die folgenden Ähnlichkeitsaspekte:

- Die strukturelle Ähnlichkeit der Objektrepräsentation.
- Die sprachliche Ähnlichkeit der Objektrepräsentation.
- Die semantische Ähnlichkeit der Objektrepräsentation.
- Funktionale Ähnlichkeitsaspekte, also abhängig vom Gebrauch der Objekte.
- Ähnliche Verarbeitung in *kognitiven Prozessen*.

SOBIK [213] stellt fest, dass ein auf der Grundlage einer formalen Repräsentation definiertes Ähnlichkeitsmaß diese unterschiedlichen Aspekte nicht alle gleichzeitig erfassen kann und dass der Ähnlichkeitsbegriff nicht vollständig formalisierbar ist. Das bedeutet aber, dass bei einem Entwurf eines Ähnlichkeitsmaßes darauf geachtet werden muss, ob das Maß Ähnlichkeitswerte erzeugt, die kognitiv plausibel und interpretierbar sind.

Ähnlichkeitsmaße sind keineswegs auf die Mathematik oder Informatik beschränkt, sie kommen in vielen anderen Wissenschaftsbereichen vor. Danach gibt es in den unterschiedlichsten wissenschaftlichen Gebieten viele Anwendungsmöglichkeiten für Ähnlichkeitsmaße, beispielsweise im Information Retrieval [81], in der Clusteranalyse [3, 78], in der Soziologie [148] und in der Psychologie [97, 233]. Je nachdem welche Objekte man aber untersucht ist die Vorgehensweise, die Ähnlichkeit zwischen den jeweiligen Objekten zu definieren, unterschiedlich. Dazu betrachte man die folgenden Beispiele aus der Mathematik:

1. Zwei Dreiecke sind in der Geometrie ähnlich, falls die folgende Aussage wahr ist: Stimmen zwei Dreiecke in zwei Winkeln überein, dann stimmen sie auch im dritten Winkel überein und sind somit per Definition ähnlich. Allgemeiner formuliert sind zwei geometrische Objekte ähnlich, wenn sie durch Drehung, Spiegelung und Streckung ineinander überführt werden können.
2. Zwei quadratische Matrizen  $(X_{ij})_{ij}$  und  $(Y_{ij})_{ij}$ ,  $1 \leq i \leq n, 1 \leq j \leq n$  heißen in der Algebra ähnlich, falls eine invertierbare Matrix  $I$  existiert, so dass die Matrizenungleichung  $X = IYI^{-1}$  gilt.

Diese Beispiele zeigen, dass es von der Beschaffenheit der Objekte abhängt, „wie“ die Ähnlichkeit definiert und verifiziert wird. Im ersten Beispiel wird die Ähnlichkeit durch *geometrische Operationen*<sup>3</sup> festgestellt, im zweiten durch Ausführung *algebraischer Operationen*<sup>4</sup>.

Im Hinblick auf eine neue Methode für die Ähnlichkeitsbestimmung strukturierter Objekte, die in Kapitel (5) mathematisch motiviert und entwickelt wird, werden in dieser Arbeit Instanzen einer speziellen Klasse von gerichteten Graphen als Objektrepräsentation betrachtet. Um sinnvolle und aussagekräftige Ähnlichkeitsmaße zu konstruieren, ist jedoch zunächst ein genaueres Verständnis der Begriffe Abstand, Distanz und Metrik hilfreich.

### 4.1.3 Abstand, Distanz und Metriken

In Kapitel (4.1.2) wurde der Ähnlichkeitsbegriff zwischen strukturierten Objekten motiviert und anhand von mathematischen Objekten erklärt, dass der Ähnlichkeitsbegriff unterschiedliche Ausprägungen hat. Je nachdem, welche Objekte man betrachtet, ist das, was man unter der Ähnlichkeit dieser Objekte verstehen will, genau zu definieren. Abstands- und Ähnlichkeitsmaße treten immer dann

---

<sup>3</sup>Z.B. Strecken und Verschieben.

<sup>4</sup>In diesem Beispiel ist es die Matrixinversion und das Ausrechnen des Matrizenproduktes.

auf, wenn Beziehungen und spezifische Eigenschaften von strukturierten Objekten beschrieben werden. Dabei hängen der Abstands- und der Ähnlichkeitsbegriff unmittelbar zusammen: Ein gemäß des zu Grunde liegenden Ähnlichkeitsmaßes hoher Ähnlichkeitswert korreliert stark mit einem kleinen Abstandswert und umgekehrt. Der bekannteste Abstandsbegriff ist der euklidische Abstand, der nach dem Geometer EUCLID benannt ist. Um weitere Abstände vorzustellen, ist es sinnvoll, den grundlegenden Begriff der Metrik zu definieren [111].

**Definition 4.1.9 (Metrik)** *Ein metrischer Raum ist ein Tupel  $(X, d)$ , bestehend aus einer nichtleeren Menge  $X$  und einer Abbildung  $d : X \times X \longrightarrow \mathbb{R}$  mit den folgenden Eigenschaften:*

- $d(x, y) \geq 0 \quad \forall x, y \in X$  und  $d(x, y) = 0 \iff x = y$  (Positivität),
- $d(x, y) = d(y, x) \quad \forall x, y \in X$  (Symmetrie),
- $d(x, y) \leq d(x, z) + d(z, y) \quad \forall x, y, z \in X$  (Dreiecksungleichung).

Die Abbildung  $d$  heißt *Metrik* auf der Menge  $X$ , und  $d(x, y)$  heißt *Abstand* zwischen den Punkten  $x$  und  $y$ .

Der bereits erwähnte euklidische Abstand ist nun für zwei Vektoren  $x, y \in \mathbb{R}^n$  definiert durch

$$d(x, y) := \left( \sum_{i=1}^n (x_i - y_i)^2 \right)^{\frac{1}{2}}. \quad (4.2)$$

In der Ebene wird der euklidische Abstand zwischen zwei Punkten  $x = (x_1, x_2)$  und  $y = (y_1, y_2)$  auch als *Luftlinienabstand* bezeichnet. Equivalent kann dieser Abstand auch in der Menge der komplexen Zahlen [184] ( $X = \mathbb{C}$ ) definiert werden. Weiter lassen sich allgemeinere Metriken definieren wie beispielsweise die bekannte MINKOWSKI Metrik,

$$d_p(x, y) := \left( \sum_{i=1}^n (x_i - y_i)^p \right)^{\frac{1}{p}}. \quad (4.3)$$

Für  $p = 2$  geht der euklidische Abstand aus der MINKOWSKI Metrik hervor.

Es existieren auch Abstandsmaße, die auf nicht euklidischen Metriken beruhen. Nicht euklidische Abstände findet man beispielsweise im POINCARÉ-schen Geometriemodell und in der Funktionentheorie [184]. Abschließend dazu sei erwähnt, dass es Abstands- und Ähnlichkeitsmaße gibt, die durch *Semimetriken* induziert werden. Bei einer Semimetrik ist die Dreiecksungleichung in Definition (4.1.9) nicht erfüllt, ansonsten gilt die Positivität und die Symmetrie. In Kapitel (5.6)



werden spezielle Abstands- und Ähnlichkeitsmaße zur Bewertung von *Sequenz-Alignments*<sup>5</sup> benötigt. Abschließend für dieses Kapitel werden die formalen Definitionen für Abstands- und Ähnlichkeitsmaße aufgeführt.

**Definition 4.1.10 (Abstands- und Ähnlichkeitsmaß)** *Es sei  $X$  eine Menge. Eine positive reelle Funktion  $\omega : X \times X \longrightarrow [0, 1]$  heißt Abstandsmaß (distance measure), falls die folgenden Bedingungen gelten:*

- $\omega(x, y) = \omega(y, x) \quad \forall x, y \in X.$
- $\omega(x, x) = 0 \quad \forall x \in X.$

*Eine positive reelle Funktion  $\pi : X \times X \longrightarrow [0, 1]$  heißt Ähnlichkeitsmaß (similarity measure), falls die folgenden Bedingungen gelten:*

- $\pi(x, y) = \pi(y, x) \quad \forall x, y \in X.$
- $\pi(x, x) = 1 \quad \forall x \in X.$

## 4.2 Strukturelle Ähnlichkeit von Graphen

Da in Kapitel (5) die Ähnlichkeit relational repräsentierter Objekte, nämlich von Graphen untersucht wird, soll nachfolgend ein kurzer Überblick über existierende Forschungsarbeiten gegeben werden, die sich mit der Bestimmung der strukturellen Ähnlichkeit von Graphen beschäftigen. In vielen Anwendungsbereichen, z.B. in der Mustererkennung oder in der Bildverarbeitung, ist es wichtig und entscheidend, Aussagen über die Ähnlichkeit der jeweiligen Objekte zu treffen. Das Kernproblem kann allgemeiner formuliert werden: Es sei  $D$  eine Datenbank, in der Objekte mit bekannter Objektrepräsentation enthalten sind. Zu einem unbekannten Objekt als Eingabe sind nun diejenigen Objekte in  $D$  gesucht, die ähnlich<sup>6</sup> zur Eingabe sind. Im Fall einer graphbasierten Repräsentation wird diese Aufgabe in der Fachliteratur allgemein als *Graphmatching* [38, 39] bezeichnet. Ist von Graphmatching die Rede, so findet man oft die Unterscheidungen *exaktes Graphmatching* und *inexaktes Graphmatching* [38, 39].

Falls zwei Graphen  $\mathcal{H}_1$  und  $\mathcal{H}_2$  gegeben sind, dann wird das exakte Graphmatching als die Aufgabe bezeichnet, den Graphisomorphismus<sup>7</sup> [8] von  $\mathcal{H}_1$  auf  $\mathcal{H}_2$

---

<sup>5</sup>Siehe Kapitel (5.4).

<sup>6</sup>Basierend auf einem zu Grunde liegenden Ähnlichkeitsmaß.

<sup>7</sup>Das Auffinden des Isomorphismus zwischen zwei Graphen wird auch allgemeiner als Graphisomorphieproblem bezeichnet.

oder den Untergraphisomorphismus<sup>8</sup> – das ist der Isomorphismus zwischen  $\mathcal{H}_1$  und einem Untergraph von  $\mathcal{H}_2$  – zu finden. Ein sehr bekannter Algorithmus zur Graphisomorphiebestimmung stammt von ULLMAN [234]. Aus Komplexitätsgründen ist das exakte Graphmatching jedoch für praktische Anwendungen, bei denen die zu Grunde liegenden Graphen von höherer Ordnung sind, kaum einsetzbar. Genauer formuliert ist das Graphisomorphieproblem von der Komplexitätsfrage her offen. Es ist zwar bisher kein effizienter Algorithmus mit *polynomialer* Laufzeit bekannt, doch konnte auch nicht die NP-Vollständigkeit [200] dieses Problems bewiesen werden. Jedoch ist die NP-Vollständigkeit für das Untergraphisomorphieproblem bewiesen. Tiefgehendere Untersuchungen über die Komplexität der Graphisomorphie findet man in [5, 24, 198]. Das inexakte Graphmatching wird in der Literatur oft als das Problem aufgefasst, eine Folge von kostenbewerteten Transformationsschritten – Einfügung, Ersetzung und Löschung von Knoten und Kanten – derart anzugeben, so dass diese Transformationsfolge einen Graph  $\mathcal{H}_1$  in den Graph  $\mathcal{H}_2$  umwandelt und die gesamten Transformationskosten der Folge minimal ausfallen.

Mit dem Ziel, die eben genannten zwei Hauptklassen des Graphmatchings zu verfeinern, lassen sich viele aus der Literatur bekannten Ansätze, die sich mit der Auffindung von Graphabständen<sup>9</sup> beschäftigen, in gemeinsame Definitionsprinzipien eingruppiieren. Diese Definitionsprinzipien sagen etwas Grundlegendes über die Idee aus, auf der ein Verfahren zur Bestimmung der strukturellen Ähnlichkeit von Graphen basiert. Bekannte Definitionsprinzipien für Graphabstände, die sich bei KADEN [131] finden, sind beispielsweise:

- Graphabstände durch eine minimale Anzahl von Änderungen.
- Graphabstände durch maximale Übereinstimmung.
- Graphabstände auf der Basis von Graphgrammatiken.
- Graphabstände, deren Definition auf verschiedenen Prinzipien beruhen [132, 133, 227].

Im Folgenden werden einige wesentliche Arbeiten vorgestellt, die sich nach diesen Definitionsprinzipien eingruppiieren lassen. Das erste Grundprinzip beruht darauf, durch eine minimale Anzahl von Änderungen – z.B. das Löschen und das Einfügen von Knoten und Kanten – einen Graph  $\mathcal{H}_1$  in einen Graph  $\mathcal{H}_2$  zu überführen. Graphabstände, die auf diesem Prinzip beruhen, wurden beispielsweise von SANFELIU et al. [188, 189] und SHAPIRO et al. [207] konstruiert. KADEN

---

<sup>8</sup>Das Auffinden des Isomorphismus zwischen Graphen und Untergraphen (Subgraphen) wird allgemeiner auch als Untergraphisomorphie- oder Subgraphisomorphieproblem bezeichnet.

<sup>9</sup>Es gilt: Je ähnlicher zwei Graphen sind, desto geringer ist ihr Abstand und umgekehrt.

hat in [130] Graphmetriken durch Graphrelationen definiert, die diesem Prinzip konzeptionell sehr nahe stehen.

Bekannte Arbeiten gibt es vor allen Dingen im Bereich der Graphabstände durch maximale Übereinstimmung. BUNKE et al. legen mit [35, 36] den Grundstein für einen wichtigen Vertreter von Graphabständen aus diesem Definitionsprinzip [131]. Es seien zwei markierte Graphen

$$\begin{aligned}\mathcal{H}_1 &:= (V_1, E_1, f_{V_1}, f_{E_1}, A_V, A_E) \\ \mathcal{H}_2 &:= (V_2, E_2, f_{V_2}, f_{E_2}, A_V, A_E)\end{aligned}$$

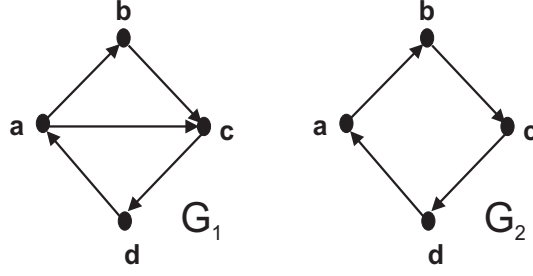
gegeben. Dann definieren BUNKE et al. ein „*Inexact Match*“ zwischen  $\mathcal{H}_1$  und  $\mathcal{H}_2$  als eine Abbildung  $m : V_1 \longrightarrow V_2 \cup \{\$ \}$ . Dabei gilt  $m(v) = m(v')$  genau dann, wenn  $m(v) = \$$  und  $m(v') = \$$ ,  $\forall v, v' \in V_1$ . Nach dieser Definition können die folgenden Fälle auftreten, falls ein „*Inexact Match*“ zwischen  $\mathcal{H}_1$  und  $\mathcal{H}_2$  besteht:

1. Der Knoten  $v \in V_1$  kann gelöscht werden.
2. Ein Knoten  $v \in V_1$  kann ersetzt werden durch einen Knoten  $v' \in V_2$ .
3. Ein Knoten  $v' \in V_2$  kann eingefügt werden.

Weiter gilt:  $v \in V_1$  wird ersetzt durch  $m(v) \in V_2$  genau dann, wenn  $m(v) = \$$ . Dabei drückt  $m(v) = \$$  die Löschung von  $v \in V_1$  und die gleichzeitige Einfügung eines Knotens  $v' \in V_2 \setminus \{m(V_1)\}$  aus.  $c(m)$  sind nun die Kosten eines „*Inexact Match*“, die durch die Summierung der Einzelkosten der eben erklärten Operationen definiert sind. Falls jetzt  $m_1, m_2, \dots, m_n$  alle theoretisch möglichen „*Inexact Matches*“ zwischen  $\mathcal{H}_1$  und  $\mathcal{H}_2$  sind, dann heißt  $m'$  ein „*Optimal Inexact Match*“, wobei die Eigenschaft  $c(m') = \min\{c(m_i) \mid 1 \leq i \leq n\}$  erfüllt sein muss. Ein „*Optimal Inexact Match*“ ist also die Transformationsfolge, die  $\mathcal{H}_1$  unter minimalen Kosten nach  $\mathcal{H}_2$  transformiert. Unter Annahme einfacher mathematischer Beziehungen bezüglich der Transformationskosten erhalten BUNKE et al. [35] das folgende wichtige Resultat:

**Satz 4.2.1** *Es seien  $d(\mathcal{H}_1, \mathcal{H}_2)$  die Kosten des „*Optimal Inexact Match*“ zwischen  $\mathcal{H}_1$  und  $\mathcal{H}_2$ . Dann ist  $d(\mathcal{H}_1, \mathcal{H}_2)$  eine Graphmetrik.*

Viele weitere Arbeiten hatten das Ziel, Graphabstände bezüglich maximaler Übereinstimmung durch größte gemeinsame isomorphe Untergraphen zu konstruieren. So definiert ZELINKA [254] erstmalig einen Graphabstand über der Menge aller *Isomorphieklassen* von einer Klasse von Graphen. Unter einer Isomorphieklasse versteht man hier die Menge derjenigen Graphen, die zu einem vorgegebenen Graphen isomorph sind. ZELINKA betrachtet dabei Graphen mit gleicher Knotenanzahl, die keine Knotenmarkierungen, keine Schlingen und keine Mehrfachkanten



**Abbildung 4.2:** Zwei knotenmarkierte gerichtete Beispielgraphen.

besitzen. Der so genannte ZELINKA-Abstand beruht darauf, dass zwei Graphen umso ähnlicher sind, je höher die Ordnung eines gemeinsamen isomorphen induzierten Untergraphen ist. Die zentrale Aussage seiner Arbeit [254] ist, dass der ZELINKA-Abstand die Eigenschaften einer Graphmetrik besitzt.

**Satz 4.2.2** *Es seien  $\mathcal{H}, \tilde{\mathcal{H}}$  unmarkierte Graphen ohne Schlingen und Mehrfachkanten. Weiter gelte  $|V| = |\tilde{V}| = n$ .  $\overline{SUB}_m(\mathcal{H})$  bezeichnet die Menge der induzierten Untergraphen der Ordnung  $m$ .  $\mathcal{H}^*$  bezeichnet die Isomorphieklasse von solchen Graphen, in der  $\mathcal{H}$  liegt. Weiterhin sei*

$$SUB_m(\mathcal{H}) := \{\mathcal{H}^* \mid \mathcal{H} \in \overline{SUB}_m(\mathcal{H})\}. \quad (4.4)$$

$SUB_m(\mathcal{H})$  ist gerade die Menge der Isomorphieklassen, in denen die induzierten Untergraphen der Ordnung  $m$  von  $\mathcal{H}$  liegen. Dann ist

$$d_Z(\mathcal{H}, \tilde{\mathcal{H}}) := n - SIM(\mathcal{H}, \tilde{\mathcal{H}}) \quad (4.5)$$

eine Graphmetrik, wobei

$$SIM(\mathcal{H}, \tilde{\mathcal{H}}) := \max\{m \mid SUB_m(\mathcal{H}) \cap SUB_m(\tilde{\mathcal{H}}) \neq \{\}\}. \quad (4.6)$$

SOBIK [213, 214] verallgemeinerte die Graphmetrik auf knoten- und kantenmarkierte Graphen beliebiger Ordnung.

**Satz 4.2.3** *Es sei  $\mathcal{H} := (V, E, f_V, f_E, A_V, A_E)$  ein endlicher markierter Graph. Sind jetzt  $\mathcal{H}, \tilde{\mathcal{H}}$  endliche, markierte Graphen beliebiger Ordnung, dann ist*

$$d_S(\mathcal{H}, \tilde{\mathcal{H}}) := \max\{|\mathcal{H}|, |\tilde{\mathcal{H}}|\} - SIM(\mathcal{H}, \tilde{\mathcal{H}}) \quad (4.7)$$

eine Graphmetrik.

Als Beispielanwendung betrachte man die Abbildung (4.2). Die direkte Anwendung von Satz (4.2.3) liefert  $d_S(G_1, G_2) = 4 - 3 = 1$ . Weiterhin führte SOBIK in [213] Graphmetriken ein, indem er das Konzept von ZELINKA, also die Basierung

auf dem größten gemeinsamen induzierten Untergraphen, ersetzt durch kleinste unterscheidender induzierter Untergraphen. Eine interessante Anwendung des ZELINKA-Abstandes lieferte KADEN [129]. Er transformierte Graphen in *Kantengraphen*<sup>10</sup>, wendete auf die transformierten Graphen den ZELINKA-Abstand an und untersuchte die Eigenschaften der transformierten Abstände. Der Kantengraph eines ungerichteten Graphen  $G = (V, E)$  ist definiert als  $\bar{G} = (E, \bar{E})$ ,  $\bar{E} := \{e, \hat{e} | e, \hat{e} \in E \text{ und } e, \hat{e} \text{ sind inzident in } G\}$ . Mit Hilfe der Graphen  $\bar{G}^m$ , die nach iterierter ( $m$ -mal) Kantengraphbildung  $\bar{G}^m$  entstehen, erhielt KADEN den

**Satz 4.2.4** *Es sei  $\Lambda_n$  die Menge der zusammenhängenden Graphen der Ordnung  $n$ . Für  $0 \leq m < n$  ist*

$$d_K^m(\mathcal{H}, \tilde{\mathcal{H}}) = d_S(\mathcal{H}^m, \tilde{\mathcal{H}}^m) = \max \{|\mathcal{H}^m|, |\tilde{\mathcal{H}}^m|\} - \text{SIM}(\mathcal{H}^m, \tilde{\mathcal{H}}^m), \quad (4.8)$$

eine Graphmetrik.

Eine weitere bekannte Arbeit aus dem Definitionsprinzip der maximalen Übereinstimmung stammt von SHAPIRO [208]. Dabei werden gerichtete Graphen  $\mathcal{H} = (V, E)$ , die zugehörige Adjazenzmatrix  $\mathcal{A}$  und die Funktion

$$f(\mathcal{H}) = \{(f(v), f(\hat{v})) | (v, \hat{v}) \in E\}$$

betrachtet, wobei  $f$  eine beliebige Permutation von  $V$  bezeichnet. Es seien nun die Graphen  $\mathcal{H}, \tilde{\mathcal{H}}$  gegeben, repräsentiert durch ihre Adjazenzmatrizen. Der Graphabstand von SHAPIRO beruht darauf, dass die Zeilen und Spalten der Adjazenzmatrix des Graphen  $\mathcal{H}$  solange permutiert werden, bis es zu einer maximalen Übereinstimmung der Matrixelemente mit der Adjazenzmatrix  $\tilde{\mathcal{A}}$  von  $\tilde{\mathcal{H}}$  kommt. Mit dieser Konstruktion erhält SHAPIRO den Graphabstand

$$d(\mathcal{H}, \tilde{\mathcal{H}}) = \min_f \|f(\mathcal{H}) - \tilde{\mathcal{A}}\|, \quad (4.9)$$

der ebenfalls eine Graphmetrik ist.

Abschließend für dieses Kapitels wird noch eine bekannte Arbeit von GERNERT vorgestellt. Genauer gesagt führte GERNERT [92] eine Methode für die Bestimmung der Ähnlichkeit zwischen Graphen, basierend auf Graphgrammatiken ein, wobei Grundlagen über Graphgrammatiken z.B. in [70, 161] zu finden sind. Ausgehend von  $S := \{G_1, G_2, \dots, G_n\}$  und unter der Bedingung, dass die Graphen  $G_i$ ,  $1 \leq i \leq n$  zusammenhängend sind, setzte GERNERT eine Graphgrammatik  $\alpha$  voraus, die eine Graphmenge  $\tilde{S} := \{\tilde{G}_1, \tilde{G}_2, \dots, \tilde{G}_p\}$ ,  $S \subseteq \tilde{S}$  erzeugt. Mit Hilfe der Funktion

$$f(\tilde{G}_i, \tilde{G}_k) := \begin{cases} 0 & : \tilde{G}_i \text{ ist isomorph zu } \tilde{G}_k \\ 1 & : \tilde{G}_i \longrightarrow \tilde{G}_k \text{ in nur einem Ersetzungsschritt} \\ \text{undefiniert} & : \text{andernfalls} \end{cases}$$

---

<sup>10</sup> Im Englischen wird der Kantengraph auch als *line graph* [8] bezeichnet.

definierte GERNERT so genannte „Pfadlängen“ zwischen den Graphen  $\tilde{G}_r$  und  $\tilde{G}_s$ , falls  $\tilde{G}_s$  durch eine gewisse Anzahl von Zwischenschritten, für die  $f$  definiert ist, ausgehend von  $\tilde{G}_r$  erzeugt werden kann. Diese Pfadlänge sei nun als  $l$  bezeichnet. Es existiert aber mindestens ein Graph  $G^*$  mit der Eigenschaft  $G^* \longrightarrow \tilde{G}_r$  und  $G^* \longrightarrow \tilde{G}_s$ . Beispielsweise könnte  $G^*$  auch der Startgraph der Graphgrammatik sein. Damit erhält GERNERT den

**Satz 4.2.5**

$$d(G_i, G_k) := \min \left\{ l(G^*, G_i) + l(G^*, G_k) \mid G^* \in \tilde{S} \wedge G^* \subseteq G_i \wedge G^* \subseteq G_k \right\}$$

ist eine Graphmetrik.

Der Satz (4.2.5) besitzt eine einfache und anschauliche Interpretation. Unter der Annahme, man hätte in Bezug auf  $G_i$  und  $G_k$  einen „optimal passenden Ursprungsgraph“  $G^\bullet$  ermittelt, setzt sich der Graphabstand aus den Transformationsschritten von  $G^\bullet \longrightarrow G_i$  und  $G^\bullet \longrightarrow G_k$  zusammen.

Abschließend sei erwähnt, dass das Graphmatching auf der Basis von Graphgrammatiken weiterführend in [34, 71, 93] untersucht wurde. Detaillierte Übersichtsartikel bezüglich Graphmatching sind in [35, 38, 39, 126] zu finden.

## 4.3 Graph Mining und weitere graphorientierte Ähnlichkeitsmaße

In Kapitel (2.2.2) wurden klassische Data Mining-Konzepte beschrieben, wobei das Data Mining die Entdeckung von Mustern und Strukturen in großen Datenbeständen zum Hauptziel hat. Das graphbasierte Data Mining, welches auch als *Graph Mining* [121, 253] bezeichnet wird, beschäftigt sich mit der Wissensexploration in graphbasierten Daten. Ausgehend von einer Datenbank graphbasierter Objekte ist die Suche [241] nach ähnlichen Graph- und Untergraphmustern innerhalb des Datenbestandes eine typische Aufgabenstellung im Graph Mining. Allgemeiner werden in [203] oft gestellte Problemstellungen des graphbasierten Data Mining skizziert:

1. Die Entdeckung ausgezeichneter Knotenmengen, z.B. hinsichtlich der Knotenzentralität<sup>11</sup>.
2. Die Entdeckung ausgezeichneter Kantenmengen, z.B. spezieller Kantenzüge, die kürzeste Wege darstellen.

---

<sup>11</sup>Siehe Kapitel (2.3.2).

3. Die Suche und Entdeckung von Graphmustern [183, 241].

4. Strukturelle Vergleiche von Graphen.

Bezüglich Punkt (4) versteht man unter strukturellen Graphvergleichen insbesondere die Bestimmung der strukturellen Ähnlichkeit zwischen den Graphen. In Kapitel (4.2) wurden im Wesentlichen klassische Methoden zur Bestimmung der Graphähnlichkeit vorgestellt, die auf Isomorphiebeziehungen beruhen. Die Ansätze, die bezogen auf Punkt (4) in diesem Kapitel (4.3) diskutiert werden, stützen sich auf Methoden der theoretischen Informatik und des maschinellen Lernens. Im ersten Fall sind damit Verfahren zur Bestimmung der Graphähnlichkeit angesprochen, die auf der Basis von sogenannten Sequenz-Alignments<sup>12</sup> die strukturelle Ähnlichkeit zwischen Graphen einer wichtigen Graphklasse – den Bäumen – beschreiben. Eine analoge Basis, die diesem Verfahren zu Grunde liegt, ist bezüglich der Problemstellung gegeben, die Distanz zwischen beliebigen Wörtern über einem gewählten Alphabet zu bestimmen [99, 190, 191]. Es sei  $A$  ein endliches, nichtleeres Alphabet gegeben und  $w_1, w_2$  sind Wörter aus  $A^*$ , der Menge der endlichen Zeichenketten über  $A$ . Es gilt ein bekannter Zusammenhang [124]: Für alle  $w_1, w_2 \in A^*$  ist der Wert der *Editierdistanz*<sup>13</sup> von  $w_1$  und  $w_2$  gleich dem Wert eines optimalen<sup>14</sup> Alignments von  $w_1$  und  $w_2$ . Unter einem Sequenz-Alignment versteht man hier die Zuordnung von Entsprechungen zwischen den Bausteinen von Wortsequenzen über einem zu Grunde liegenden Alphabet. Die eben beschriebene Äquivalenz lässt sich jedoch nicht für Alignments von Bäumen formulieren [124]. Aus einem Alignment von zwei Bäumen kann zwar die entsprechende Folge der nötigen Editieroperationen<sup>15</sup> konstruiert werden, die umgekehrte Reihenfolge gilt aber nicht notwendigerweise.

Bekannte Verallgemeinerungen von Sequenz-Alignments beschreiben TAI [226] und SELKOW [205]. Zum Beispiel betrachtet TAI Alignments von geordneten und ungerichteten Wurzelbäumen. Bei einem geordneten Wurzelbaum ist die Reihenfolge der Kinder  $v_1, v_2, \dots, v_{\delta(v)}$  eines Knotens  $v$  signifikant, wobei  $\delta(v)$  den Grad<sup>16</sup> von  $v$  bezeichnet. Algorithmische Verbesserungen dieser Variante und problem- bzw. anwendungsorientierte Weiterentwicklungen erfolgen in [112, 209, 255, 256], die besonders für Problemstellungen innerhalb der *Bioinformatik* [145] genutzt werden. So beschreiben HÖCHSTMANN et al. [112] ein Verfahren, welches auf strukturellen Baumvergleichen beruht, um lokal und global ähnliche Baummuster in *RNA-Sekundärstrukturen* [145] zu bestimmen. Dabei sind genetische Informationen von Organismen in den meisten Fällen in der DNA gespeichert.

---

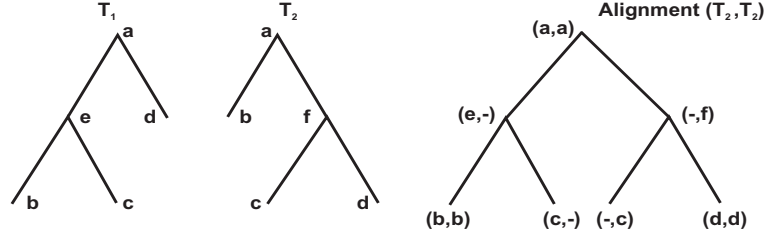
<sup>12</sup>Siehe auch Definition (5.4.1) in Kapitel (5.4).

<sup>13</sup>Siehe Kapitel (5.5), Gleichung (5.14).

<sup>14</sup>Das heißt unter minimalen Kosten bezüglich der Alignmentbewertung. Siehe auch Kapitel (5.5).

<sup>15</sup>Siehe Kapitel (5.5).

<sup>16</sup>In Kapitel (5.2) wird der Grad für gerichtete und ungerichtete Graphen formal definiert.



**Abbildung 4.3:** Zwei knotenmarkierte, geordnete und ungerichtete Wurzelbäume  $T_1$  und  $T_2$ . Das rechte Bild zeigt ein optimales Alignment von  $T_1$  und  $T_2$ .

Um solche Informationen nutzbar zu machen, müssen sie in Proteine übersetzt werden. Als Zwischenstufe einer solchen Übersetzung kann die RNA betrachtet werden. Neben einer beschreibenden RNA-Sequenz, der Primärstruktur, existiert in den meisten Fällen die RNA-Sekundärstruktur, die ausdrückt, welche Basen miteinander gepaart sind [145]. Dabei werden in [112] zunächst die ringförmigen RNA-Sekundärstrukturen in geordnete und ungerichtete Wurzelbäume transformiert. Um strukturelle Vergleiche von diesen Bäumen vorzunehmen, verwenden HÖCHSTMANN et al. in [112] das entsprechende Verfahren von JIANG et al. [124] und verallgemeinern es zur Bestimmung der strukturellen Ähnlichkeit von *Wäldern*. Dabei versteht man in [112] unter einem *Wald* eine Sequenz aus den betrachteten Bäumen. Um das ursprüngliche Verfahren von JIANG et al. detaillierter zu beschreiben, wird zunächst die Knoteneinfügung und die Knotenlöschung bezüglich eines geordneten und ungerichteten Wurzelbaums  $T = (V_T, E_T)$  erklärt. Die Einfügung (insert) eines Knotens  $u \in V_T$  am Knoten  $v \in V_T$  bedeutet [226]: Alle oder eine Teilmenge der Kinder von  $v$  werden Kinder von  $u$  und  $v$  wird Vaterknoten von  $u$ . Die Löschung (deletion) eines Knotens  $u \in V_T$  ist gerade komplementär zur Einfügung und bedarf daher keiner separaten Definition [226]. Es seien nun  $T_1$  und  $T_2$  knotenmarkierte, geordnete und ungerichtete Wurzelbäume. Ein Alignment von  $T_1$  und  $T_2$  erhält man auf der Basis der folgenden Schritte [124]:

1. Füge das Lückensymbol '-' solange in  $T_1$  und  $T_2$  ein, bis die dadurch entstehenden Bäume  $\tilde{T}_1$  und  $\tilde{T}_2$ , abgesehen von den Knotenmarkierungen, dieselbe Struktur besitzen.
2. Lege die Bäume  $\tilde{T}_1$  und  $\tilde{T}_2$  bildlich übereinander.
3. Den Wert des Alignments von  $T_1$  und  $T_2$  erhält man dadurch, indem die Kostenwerte aller Paare von Knotenmarkierungen aufsummiert werden.
4. Das optimale Alignment von  $T_1$  und  $T_2$  zeichnet sich durch minimale Kosten unter allen theoretisch möglichen Alignments aus.

Die Abbildung (4.3) [124] zeigt beispielhaft ein optimales Alignment zweier Bäume  $T_1$  und  $T_2$ . Auf der Basis dieses Verfahrens berechnen HÖCHSTMANN et al.



[112] mit Hilfe der *dynamischen Programmierung*<sup>17</sup> die globale Ähnlichkeit von Wäldern  $W_1$  und  $W_2$ , die durch die maximale Summe der Knotenmarkierungen eines Alignments von  $W_1$  und  $W_2$  gegeben ist. Weiterhin geben sie Erweiterungen an, um auf Grundlage der Methode von JIANG et al. maximal ähnliche Teilbäume in Bäumen oder Wäldern zu bestimmen. Ein Problem während der Berechnung eines Baum-Alignments ist, dass oft nicht unmittelbar klar ist, wie z.B. Übereinstimmungen und Lückenpaarungen passend bewertet werden sollen. WANG et al. [240] geben daraufhin parametrische Algorithmen zur Bestimmung der strukturellen Ähnlichkeit von geordneten Bäumen an. Genauer entwickeln sie Alignment-Techniken für geordnete Bäume, einmal ohne und im anderen Fall mit Lückenstrafe. Ziel der Untersuchungen von WANG et al. ist es, Parameterintervalle zu finden, so dass in jedem solchen Intervall optimale Alignments existieren. Eine weitere Methode zur Bestimmung der strukturellen Ähnlichkeit von Bäumen auf der Grundlage eines abstrakten Maßes, sowie weitere Arbeiten in diesem Problemkreis, sind bei OOMMEN et al. [170] zu finden. Als letzte Arbeit in diesem Ideenkreis sei eine Arbeit von MEHLER [154] erwähnt. MEHLER transformiert in [154] Texte in sogenannte „*Text Structure Strings*“ [154], welche die Struktur der Texte, z.B. Sektionen, Paragraphen und Satzebene, widerspiegeln. Dabei stellen diese Strukturen Wurzelbäume dar. Um nun die Strukturen zu vergleichen, wendet MEHLER die bekannte LEVESTEIN-Metrik<sup>18</sup> auf die zu Grunde liegenden „Text Structure Strings“ an.

Ein Graph Mining-Ansatz auf der Grundlage des maschinellen Lernens stellen HORVÁTH et al. [115] vor. Ziel dieser Arbeit ist es, auf der Grundlage der bekannten NCI-HVI-Datensammlung [115], mit Hilfe einer Support Vector Machine, Moleküle in Form von ungerichteten Graphen anhand bekannter Trainingsbeispiele zu lernen. Es handelt sich dabei um ein überwachtetes Lernverfahren, welches anhand von graphbasierten Trainingsbeispielen, für die eine auszeichnende Grapheneigenschaft bekannt ist, eine unbekannte Funktion mit minimalem Vorhersagefehler lernt. Auf diese Weise können charakteristische Molekülstrukturen im Bereich der HIV-Forschung in einer großen Datenmenge identifiziert werden. Ein entscheidender Kernpunkt dieses Verfahrens ist jedoch die Definition von geeigneten und effizienten Kernel-Funktionen<sup>19</sup>  $k : \mathcal{G} \times \mathcal{G} \rightarrow \mathbb{R}$  [90], die die Ähnlichkeit zwischen den gelabelten graphbasierten Instanzen aus  $\mathcal{G}$  detektieren. Dabei basieren bekannte Graph-Kernels, z.B. [27] auf dem Prinzip, die Häufigkeit der in den Graphen vorkommenden Untergraphen zu bestimmen und anschließend die auszeichnende Eigenschaft der Kernel-Funktion auf diese Untergraphmengen anzuwenden. HORVÁTH et al. schlagen in [115] einen Graph-Kernel vor, der auf Teilmengen von zyklischen und baumartigen Graphmustern basiert, wobei auch Permutationen von Zyklen mit einbezogen wurden. Dabei werden alle Gra-

---

<sup>17</sup>Siehe Kapitel (5.5).

<sup>18</sup>Siehe Kapitel (5.5).

<sup>19</sup>Siehe Kapitel (3.3).

phen der zu Grunde liegenden Datenmenge auf diese Teilmengen abgebildet, unabhängig davon, wie häufig diese Muster auftreten. Der Grundmechanismus, der dieser Kernel-Funktion zu Grunde liegt, basiert auf den folgenden allgemeineren Teilschritten [115, 116]:

1. Zerlegung eines komplexen Graphobjektes in eine Menge von charakteristischen Graphmustern.
2. Berechnung der Schnittmenge zweier Mengen von Graphmustern.

Gemäß dieser Schritte wurde die Kernel-Funktion  $k_{CP}(G_i, G_j)$  (CP=Cyclic Patterns) zweier graphbasierter Repräsentationen  $G_i, G_j$  schließlich definiert als

$$k_{CP}(G_i, G_j) := |C(G_i) \cap C(G_j)| + |\tau(G_i) \cap \tau(G_j)|,$$

wobei  $C(G)$  bzw.  $\tau(G)$  die Menge der zyklischen Muster bzw. der Baummuster von  $G$  bezeichnet. In [115] wurde jedoch gezeigt, dass die Berechnung der Kernel-Funktion  $k_{CP}(G_i, G_j)$ , die auf der Zerlegung der Graphobjekte in zyklische und baumartige Untergraphen basiert, exponentielle Laufzeit erfordern kann. Um diesem negativen Aspekt entgegenzuwirken, betrachten HORVÁTH et al. [115] die Menge der einfachen Zyklen, wobei der Weg

$$z = \{v_0, v_1\}, \{v_1, v_2\}, \dots, \{v_{k-1}, v_k\} \quad \text{mit} \quad v_0 = v_k, v_i \neq v_j, 1 \leq i < j \leq k$$

einen einfachen Zyklus darstellt. Darauf basierend wird in [115] eine Zahl effizient berechnet, die die Anzahl der einfachen Zyklen eines Graphen  $G$  nach oben beschränkt. Dieses Vorgehen stützt sich wiederum auf die Annahme, dass im Hinblick auf die zu Grunde liegende Datenmenge die Anzahl der einfachen Zyklen durch eine feste Zahl beschränkt ist. Somit lassen sich auf Basis der Menge der einfachen Zyklen einerseits zyklische und andererseits Graphmuster in Form von Bäumen extrahieren. Mit Hilfe der bereits erklärten Durchschnittsbildung werden damit effizient berechenbare Kernel-Funktionen definiert. Damit erreichen HORVÁTH et al. schließlich bessere Ergebnisse der Performancemessung, als mit herkömmlichen Kernel-Funktionen.

Eine effiziente Methode zur Klassifikation großer ungerichteter Graphen im Bereich des unüberwachten Lernens, die auf dem Graphähnlichkeitsmodell aus Kapitel (5) beruht, entwickelten EMMERT-STREIB et al. [73]. Dabei beruht diese Methode auf dem Prinzip, welches auch in der geschilderten Arbeit von HORVÁTH et al. [115] angewendet wurde:

*Die Dekomposition komplexer Graphobjekte in Teilmengen bekannter Graphmuster, z.B. Bäume, wobei die Strukturen dieser Teilmengen nun leichter und effizienter zu verarbeiten sind als die Ursprungsgraphen.*

Auf der Grundlage einer eindeutigen Dekompositionsmethode für ungerichtete Graphen, zerlegen EMMERT-STREIB et al. diese Graphen in ihre zugehörigen Mengen von hierarchisierten und gerichteten Graphen<sup>20</sup>. Die Frage der strukturellen Ähnlichkeit ungerichteter Graphen kann mit Hilfe einer binären Graphklassifikationsmethode beantwortet werden: Zwei ungerichtete Graphen  $G_1$  und  $G_2$  sind genau dann ähnlich, wenn die Ähnlichkeitswertverteilungen der zugehörigen Mengen hierarchisierter und gerichteter Graphen ähnlich sind. Damit kann nun für die Graphklassifikation die wesentliche Frage beantwortet werden, ob  $G_1$  und  $G_2$  derselben Graphklasse angehören. Hinsichtlich bekannter Arbeiten, z.B. [169, 172] zur Berechnung der strukturellen Ähnlichkeit großer ungerichteter Graphen im Hinblick auf die Klassifikation, ist der geschilderte Ansatz von EMMERT-STREIB et al. neuartig, da ganzheitliche Graphvergleiche und keine Graphmuster, z.B. Zusammenhangskomponenten oder einfache strukturelle Kennzahlen zur Klassifikation herangezogen werden. Das Anwendungsgebiet dieses Klassifikationsverfahrens wird in Kapitel (7.2) motiviert und detaillierter dargestellt.

Da das Hauptziel dieser Arbeit in der Entwicklung von neuen ähnlichkeitsbasierten Analysemethoden graphbasierter Dokumente besteht, werden nun bekannte Verfahren beschrieben, die sich speziell mit der Bestimmung der Ähnlichkeit web-basierter Dokumentstrukturen befassen. Beispielsweise stellt BUTTLER [41] in seiner Übersichtsarbeit Ansätze zur Bestimmung der strukturellen Ähnlichkeit web-basierter Dokumente vor, wobei es sich bei den betrachteten Dokumenten um XML oder DOM-Strukturen [45] handelt. Die Ansätze lassen sich in charakteristische Gruppen einteilen:

1. Ähnlichkeitsmaße, die auf dem Editiermodell von Bäumen basieren. Sie wurden zu Anfang dieses Kapitels (4.3) besprochen, wobei ein typischer Vertreter die bereits diskutierte Arbeit von JIANG et al. [124] ist.
2. Ähnlichkeitsmaße, die auf der Häufigkeit von Tags beruhen.
3. Ähnlichkeitsmaße, die auf der *Fouriertransformation* beruhen.
4. Bewertung der Dokumentähnlichkeit auf der Grundlage der Ähnlichkeit von Pfaden.

Wie bereits in diesem Kapitel (4.3) erwähnt, ist die Berechnung der Editierdistanz<sup>21</sup>  $d$  in Bezug auf die Bestimmung eines optimalen Alignments wesentlich. Im Bereich der Sequenz-Alignments von Wörtern über einem beliebigen Alphabet besteht sogar die bekannte Äquivalenz [99, 124] zwischen Editierdistanz und

---

<sup>20</sup>Zur formalen Definition hierarchisierter und gerichteter Graphen siehe Definition (5.3.1) in Kapitel (5.3).

<sup>21</sup>Siehe Kapitel (5.5), Gleichung (5.14).

Alignment. Auf Grundlage der Editierdistanz zweier Dokumentstrukturen  $D_i$  und  $D_j$  in Form von Bäumen, definiert BUTTLER das Maß

$$\text{TEDS}(D_i, D_j) := \frac{d(D_i, D_j)}{\max(|V_i|, |V_j|)} \quad (\text{Tree Edit Distance Similarity}),$$

wobei  $V_i$  bzw.  $V_j$  die Knotenmengen der entsprechenden Baumrepräsentationen  $D_i$  bzw.  $D_j$  bezeichnen.

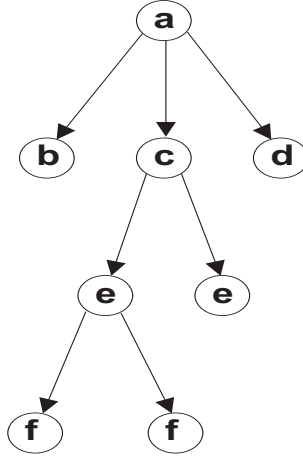
Die Bestimmung der strukturellen Ähnlichkeit graphbasierter Dokumente auf der Basis von Tags, auf die sich der Punkt (2) der obigen Aufzählung bezieht, ist die einfachste Form zur Berechnung der strukturellen Dokumentähnlichkeit. Dieser Ansatz ist naheliegend, da in XML und DOM-Strukturen das Strukturschema einer Webseite im Wesentlichen durch die Menge der Tags bestimmt wird. Es seien mit  $TG_i$  und  $TG_j$  die Tag-Mengen der Dokumentrepräsentationen  $D_i$  und  $D_j$  bezeichnet. Dann definiert BUTTLER die strukturelle Ähnlichkeit von  $D_i$  und  $D_j$  in der grundlegendsten Form als

$$\text{TS}(D_i, D_j) := \frac{|TG_i \cap TG_j|}{|TG_i \cup TG_j|} \quad (\text{Tag Similarity}).$$

Weiterhin beschreibt BUTTLER in [41] das Problem, dass unterschiedliche Webseiten, die eigentlich dasselbe Strukturschema ausdrücken, stark unterschiedliche Tag-Anzahlen besitzen können. Aus diesem Grund sind gewichtete Tag-Ähnlichkeitsmaße sinnvoll, die ebenfalls in [41] definiert werden.

Bezogen auf den Punkt (3) der Aufzählung stellt BUTTLER ein auf der Fouriertransformation [82] basierendes Verfahren vor, welches ursprünglich von FLESCA et al. [84] stammt. Hier werden lediglich die Konstruktionsschritte dieses Verfahrens kurz erläutert:

- Entfernung aller überflüssiger Informationen innerhalb des Dokuments, so dass eine Struktur verbleibt, die als Gerüst der Dokumentstruktur interpretiert werden kann. Dabei werden die jeweiligen Start- und End-Tags mit positiven ganzen Zahlen markiert.
- Transformation dieser Struktur in eine Zahlensequenz und anschließende Interpretation als *Zeitreihe*, wobei die Zeitreihe eine zeitlich geordnete Abfolge von Beobachtungen darstellt.
- Aus diesen Daten werden mit Hilfe der Fouriertransformation Signale hergestellt. Der Abstand zweier Dokumente reduziert sich auf die Berechnung der Differenz zweier Signalstärken, die durch Fouriertransformation erzeugt wurden.



**Abbildung 4.4:** Ein fiktiver DOM-Tree mit seinen syntaktischen Knotensequenzen: a/b, a/c/e/f, a/c/e/f, a/c/e/f, a/c/e und a/d.

Bezüglich Punkt (4) erläutert BUTTLER in [41] das Prinzip, die Dokumentähnlichkeit auf der Basis der Ähnlichkeit von Pfaden der zu Grunde liegenden graph-basierten Strukturen zu beschreiben. XML und HTML-Seiten, repräsentiert als DOM-Strukturen, können nämlich leicht als Sequenzen von Pfaden dargestellt werden, die jeweils von der Wurzel zu den Blättern im Baum führen. Ausführlicher wird dieses Prinzip von JOSHI et al. [127] diskutiert, indem sie das *Bag of Tree Paths*-Modell einführen. Basierend auf der baumartigen DOM-Struktur, beschreiben JOSHI et al. alle Pfade von der Wurzel bis zu den Blättern als syntaktische Knotensequenzen. Da es sich um knotenmarkierte Bäume handelt, wobei die Knotenmarkierungen XML oder HTML-Tags darstellen, sind die Pfadsequenzen als die entsprechenden Knotensequenzen zu interpretieren.

Die Abbildung (4.4) zeigt einen beispielhaften DOM-Tree mit seinen Knotensequenzen, die Pfade repräsentieren. Es sei  $D := \{d_1, d_2, \dots, d_n\}$  eine Menge bestehend aus  $n$  web-basierten Dokumenten, die durch ihre DOM-Trees dargestellt sind. Ähnlich wie bei Textklassifikationsproblemen führen JOSHI et al. eine Feature-Selektion bezüglich  $D$  durch, um alle Pfade, die nur in sehr wenigen Dokumenten vorkommen, zu entfernen. Weiterhin sei nun  $P := \{p_1, p_2, \dots, p_\rho\}$  die verbleibende Gesamtmenge der Pfade,  $f_j(p_i)$  bezeichnet die Vorkommenshäufigkeit von Pfad  $p_i$  in Dokument  $d_j$  und es gelte  $f_{max} := \max_{i,j} f_j(p_i)$ . Auf Grundlage dieser Voraussetzungen definieren JOSHI et al. die strukturelle Ähnlichkeit zwischen den Dokumenten  $d_j$  und  $d_l$  als

$$\text{SIM}(d_i, d_l) := \frac{\sum_{k=1}^{\rho} \min(d_{jk}, d_{lk})}{\sum_{k=1}^{\rho} \max(d_{jk}, d_{lk})},$$

wobei  $d_{jk} := \frac{f_j(p_k)}{f_{max}}$ . Dabei werden in diesem Modell die Knotenbeziehungen der Art Vaterknoten/Kindknoten berücksichtigt, nicht jedoch die Beziehungen der

Kindknoten untereinander. Auf dieser Vorgehensweise basierend, definieren JOSHI et al. das weitergehende *Bag of XPath*s-Modell, welches zusätzlich Beziehungen der Kindknoten ausdrücken kann.

Abschließend für dieses Kapitel wird die Überblicksarbeit von CRUZ et al. [57] angegeben. Neben den bereits schon erwähnten Methoden wie z.B. der Bestimmung der strukturellen Ähnlichkeit auf der Basis der Tag-Häufigkeiten und der Editierdistanz<sup>22</sup> von Bäumen, sei an dieser Stelle noch ein Verfahren aus [57] angesprochen, welches auf der freien Definition von Formeln basiert. Ausgehend von gegebenen Dokumentstrukturen wird zunächst eine geeignete Datenrepräsentation gesucht. Auf Grundlage von Funktionen, die Eigenschaften der transformierten Dokumentstrukturen beschreiben, können nun darauf basierend Ähnlichkeitsmaße in Gestalt von Formeln definiert werden. Je nach ausgewählter Datenrepräsentation können damit bestimmte Eigenschaften und Ausprägungen der Dokumente mit gezielter Definition von Formeln betont werden.

## 4.4 Zusammenfassende Bewertung

In den Kapiteln (4.2), (4.3) wurden im Hinblick auf das neue Graphähnlichkeitsmodell bekannte Methoden zur Bestimmung der Ähnlichkeit von Graphen und web-basierten Dokumentstrukturen vorgestellt, um die neuen Entwicklungen aus Kapitel (5) besser einordnen zu können. Abbildung (4.5) fasst die wesentlichen Ergebnisse bewertend zusammen.

Da nun in Kapitel (5) die Entwicklung eines Graphähnlichkeitsmodells für das web-basierte Graphmatching fokussiert wird, werden zunächst zwei Bedingungen angegeben, die von einem sinnvollen Graphmatching-Verfahren erfüllt werden müssen:

1. Die Verarbeitung von Massendaten.
2. Die Verarbeitung von Graphen hoher Ordnung.<sup>23</sup>

Bezüglich dieser Bedingungen ist das z.B. in Kapitel (4.2) vorgestellte exakte Graphmatching nicht für das web-basierte Graphmatching geeignet, weil einerseits die betrachteten Graphen oft unterschiedliche Ordnungen besitzen. Andererseits ist die Berechnung der Untergraphisomorphie nicht handhabbar, da die Algorithmen im schlechtesten Fall exponentielle Laufzeit besitzen und daher isomorphe Untergraphen höherer Ordnung nur in einem unrealistischen Zeitaufwand

---

<sup>22</sup>Siehe Kapitel (5.5), Gleichung (5.14).

<sup>23</sup>Denkbar wäre hier die Forderung  $1 \leq |V| \leq 10000$ .

Themenbereich	Literaturangaben	Positiv/Negativ
Graphabstände - Definitionsprinzip: Maximale Übereinstimmung (Graphisomorphie)	[129, 213, 214, 254]	Gute theoretische Fundierung/Un- genügendes Komplexitätsverhalten
Graphabstände - Definitionsprinzip: Maximale Übereinstimmung (Graphtransformatio- nen)	[35, 36]	Gute theoretische Fundierung/Un- genügendes Komplexitätsverhalten
Graphmatching basierend auf Graphgrammatiken	[34, 71, 92, 93]	Gute theoretische Fundierung/Schwie- rige Konstruktion der Graphgrammatik
Ähnlichkeitsmaße für Bäume	[112, 124, 226, 170, 240]	Effizient/Lediglich auf Bäumen definiert; Teilweise trivale Alignment-Technik
Graphmatching basierend auf Kernelmethoden	[27, 90, 115, 116]	Für spezielle Graphklassen effizient/Schwierige Konstruktion des Graphkernels
Strukturelle Ähnlichkeit web-basierter Dokumente	[41, 57, 84, 127]	Effizient/Lediglich auf DOM-Strukturen definiert; Mangelnde Strukturerfassung

**Abbildung 4.5:** Zusammenfassung der Ergebnisse aus den Kapiteln (4.2), (4.3).

zu berechnen sind. Aus dem gleichen Grund sind die Graphmetriken, die auf den größten, gemeinsamen und isomorphen Untergraphen beruhen, im Hinblick auf das web-basierte Graphmatching nicht nutzbar. Da die Anzahl der Isomorphieklassen für Graphen höherer Ordnungen unüberschaubar ist, wäre ein extrem hoher kombinatorischer Aufwand erforderlich, um solche Maße in Anwendungen einzusetzen, in denen die sofortige Berechnung der Graphähnlichkeit gefordert ist. Um eine bessere Vorstellung über die Mächtigkeit der Isomorphieklassen zu bekommen, ist in Abbildung (4.6) [107] ein Ausschnitt der Größenordnungen am Beispiel ungerichteter Graphen dargestellt. Auf Grund des ungenügenden Komplexitätsverhaltens ist diese Klasse von Verfahren zur Messung der strukturellen Ähnlichkeit für das web-basierte Graphmatching nicht einsetzbar.

Graphabstände, basierend auf Graphgrammatiken, sind ebenfalls hauptsächlich nur von theoretischem Interesse, da in der Praxis die zu Grunde liegende Graphgrammatik schwer zu bestimmen ist. Die meisten der in Kapitel (4.2) dargestellten Arbeiten wurden jedoch theoretisch intensiv untersucht und sind deshalb gut

$ V $	Anzahl der paarweise nichtisomorphen Graphen
1	1
2	2
3	4
4	11
5	34
6	156
7	1 044
8	12 346
9	271 346
10	12 005 108
11	1 018 997 864
12	165 091 172 592

**Abbildung 4.6:** Menge der Isomorphieklassen für ungerichtete Graphen.

voneinander abgrenzbar.

In Kapitel (4.3) wurden zuerst Verfahren zur Bestimmung der strukturellen Ähnlichkeit auf der Basis von Alignments vorgestellt, deren Anwendung allerdings auf Bäume beschränkt ist. Im Hinblick auf das web-basierte Graphmatching ist die Verarbeitung hierarchisierter und gerichteter Graphen von zentraler Bedeutung. Daher sind die diskutierten Alignment-Verfahren nicht anwendbar, da hierarchisierte und gerichtete Graphen eine wesentlich komplexere Kantenstruktur als reine Wurzelbäume besitzen. Im Gegensatz zu herkömmlichen Sequenz-Alignments von Wörtern beschreibt die Arbeit von HÖCHSTMANN et al. [112], die sich auf das Alignment-Verfahren von JIANG et al. [124] stützt, den Trivialfall eines Alignments. Das Alignment zwischen den Wurzelbäumen  $T_1$  und  $T_2$  besteht lediglich aus der Herstellung der maximalen strukturellen Übereinstimmung, abgesehen von Knotenmarkierungen.

HORVÁTH et al. stellen in [115] ein überwachtes Lernverfahren auf der Basis eines SVM-Klassifikators vor. Daher ist die Erzeugung von Trainingsbeispielen erforderlich, die mit den Klassen-Labels (+1) und (-1) gekennzeichnet sind. Kernel-Funktionen, die einerseits auf der Dekomposition der betrachteten Graphen in strukturell charakteristische Teilmengen und andererseits auf der Schnittmengenbildung dieser Teilmengen beruhen, sind im Allgemeinen schwierig zu verwenden. Die Vorschrift zur effizienten Berechnung solcher Kernelfunktionen liegt bezüglich realer Graphmengen nicht unmittelbar auf der Hand.

Die Methoden aus Kapitel (4.3), die sich insbesondere mit der Bestimmung der strukturellen Ähnlichkeit web-basierter Dokumente befassen, sind ebenfalls nicht auf hierarchisierte und gerichtete Graphen anwendbar. Neben den vorgestellten



Verfahren von BUTTLER [41], die auf der Editierdistanz von Bäumen beruhen, wurden in [41] auch Tag-basierte Verfahren vorgestellt. Die Vertauschung von Tags, die Strukturveränderungen in der DOM-Struktur zur Folge haben, spiegelt sich nicht in der Berechnung des Ähnlichkeitswertes wieder. Damit ist diese Klasse von Maßen zur Ähnlichkeitsmessung komplexer Dokumentstrukturen unzureichend.

Der Ansatz von JOSHI et al. [127], der die Dokumentähnlichkeit auf der Basis der Ähnlichkeit von Pfaden beschreibt, ist wieder nur auf Wurzelbäumen definiert. Dieses Verfahren nimmt zwar Bezug auf die Baumstruktur der Webseite, die Betonung unterschiedlicher struktureller Aspekte während der Ähnlichkeitsmessung, z.B. die stärkere Berücksichtigung von Eingangs- und Ausgangsgraden<sup>24</sup>, ist jedoch nicht möglich. Auf Grund der Tatsache, dass das Verfahren nur auf der Wurzelbaumstruktur operiert, kann lediglich die Ähnlichkeit web-basierter Dokumente in Form von DOM-Strukturen gemessen werden.

---

<sup>24</sup>Siehe Kapitel (5.2), Definition (5.2.1).



# Kapitel 5

## Graphbasierte Analyse und Retrieval: Neuer Ansatz

Die Bestimmung der strukturellen Ähnlichkeit von Graphen stellt ein herausforderndes Problem dar. Besonders bei ähnlichkeitsbasierten Graphanalysen auf großen Datenbeständen, wobei die Graphen von höherer Ordnung sind, ist die Konstruktion von effizienten und aussagekräftigen Ähnlichkeitsmaßen schwer. Im vorliegenden Kapitel (5) wird nun die Motivation und mathematische Modellierung einer neuen Methode zur effizienten Bestimmung der strukturellen Ähnlichkeit hierarchisierter und gerichteter Graphen angegeben. Sie ist auf Grund ihrer Konzeption für das web-basierte Graphmatching hinsichtlich Massendaten geeignet. Zum einen besteht das Hauptziel dieses neuen Ansatzes in der Umgehung von graphentheoretischen Modellen, die auf Isomorphie- oder Untergraphisomorphiebeziehungen aufbauen. Zum anderen wird die Entwicklung eines unüberwachten und parametrischen Verfahrens angestrebt, welches die strukturelle Ähnlichkeit auf der Basis ganzheitlicher Graphvergleiche bestimmt. Kapitel (5.1) stellt zunächst die grundlegende Motivation aus anwendungsorientierter und mathematischer Sicht dar. Ausgehend von weiterführenden graphentheoretischen Begriffen und Konstruktionen, die in den Kapiteln (5.2), (5.3) definiert werden, diskutiert Kapitel (5.4) den zentralen Lösungsansatz. Da das neue Verfahren auf einem Algorithmus basiert, welcher auf dynamischer Programmierung beruht, werden die erforderlichen Hilfsmittel in Kapitel (5.5) eingeführt. Mit der eigentlichen Konstruktion der Graphähnlichkeitsmaße in Kapitel (5.6) und einem experimentellen Teil in Kapitel (5.8) schließt das Kapitel (5) ab.

## 5.1 Motivation

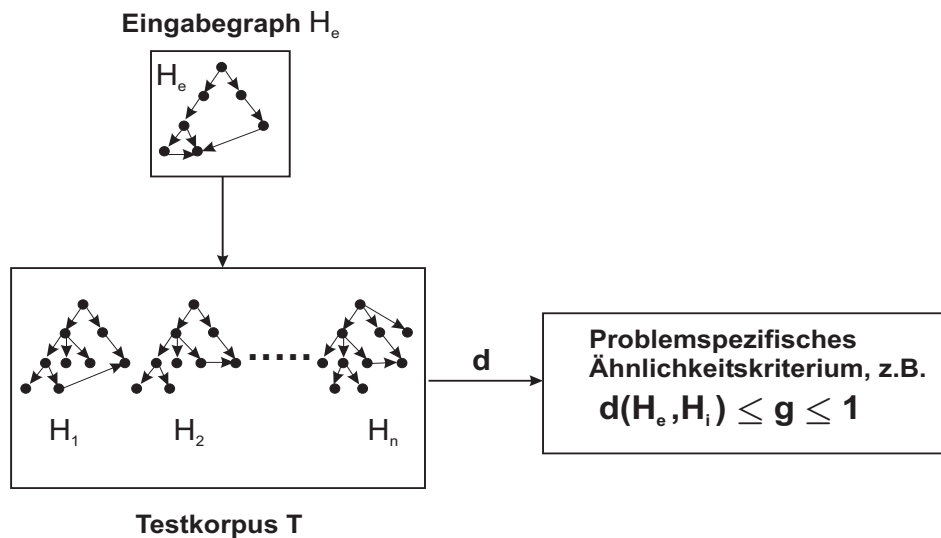
In Kapitel (4.3) wurden bekannte Arbeiten vorgestellt, die auf der Basis von Heuristiken und im Hinblick auf spezielle Graphklassen, z.B. Bäume, Methoden zur Bestimmung der Graphähnlichkeit untersuchen [112, 124, 226, 170, 240]. Allgemeiner wurden in Kapitel (4.2) Verfahren zur Bestimmung der Graphähnlichkeit besprochen. Die kritische Diskussion in Kapitel (4.4) zeigte jedoch deutlich, dass diese Verfahren, vor allem wegen der zu Grunde liegenden Isomorphiebeziehungen, für praktische Anwendungen im Web Structure Mining nicht einsetzbar sind.

Im Hinblick auf das web-basierte Graphmatching muss nun ein Verfahren gefunden werden, das im Hinblick auf Massendaten effizient und mit möglichst geringem Strukturverlust arbeitet. Um als Motivation die generelle Konstruktionsidee für ein solches Verfahren allgemeiner zu erläutern, wird im Folgenden eine Objektmenge  $O$  vorausgesetzt. Angenommen, es bestünde nun die Aufgabe, die Ähnlichkeit zwischen allen Objekten  $O_i \in O$ ,  $1 \leq i \leq |O|$  auf Grundlage einer Methode  $M_O$  zu bestimmen. Dann besteht eine wesentliche Konstruktionsidee aus den folgenden Schritten:

1. Transformation der Objekte  $O_i$  mittels einer Abbildung  $T : O \longrightarrow N$  in einen niedrigdimensionaleren Objektraum  $N$ . Dabei besitzt jedes Objekt  $O_i \in O$  eine auf der Transformation  $T$  beruhende Entsprechung  $N_i \in N$ .
2. Die Anwendung von neu definierten oder bekannten Methoden  $M_N$  zur Bestimmung der Ähnlichkeit der transformierten Objekte  $N_i$ . Dies geschieht in der Hoffnung, die Ähnlichkeit zwischen allen Objekten  $N_i$  mit minimalem Strukturverlust nun wesentlich effizienter zu bestimmen.

Für die spezielle Problemstellung mit einer konkret vorgegebenen Objektmenge  $O$  ist dabei das Auffinden einer strukturerhaltenden Abbildung  $T$  das Hauptproblem. Auf Basis dieser Konstruktionsschritte erfolgt in Kapitel (5.4) der erste Definitionsschritt des neuen Ansatzes. Dieser geschieht konkret durch Transformation der graphbasierten Objekte in eindimensionale Strukturen.

Da in dieser Arbeit Problemstellungen des Web Structure Mining, insbesondere das web-basierte Graphmatching und dessen weiterführende Anwendungen im Vordergrund stehen, muss ein solches Verfahren auch praktische Anforderungen erfüllen. Im Mittelpunkt des Verfahrens steht dann ein aussagekräftiges Ähnlichkeitsmaß, mit dem die strukturelle Ähnlichkeit graphbasierter Hypertextrepräsentationen numerisch bestimmt wird. Im Hinblick auf die geplante Anwendung sind die folgenden Bedingungen von Bedeutung:



**Abbildung 5.1:** Das web-basierte Graphmatching.

- Das Ähnlichkeitsmaß muss in seiner algorithmischen Umsetzung möglichst effizient sein, da Massendaten und Graphen höherer Ordnung verarbeitet werden.
- Das Ähnlichkeitsmaß sollte einfach mathematisch formalisierbar sein.
- Das Ähnlichkeitsmaß sollte soviel „Graphstruktur“ wie möglich erfassen. Eine Gewichtung und Bewertung spezifischer struktureller Eigenschaften auf der Basis von Parametern ist wünschenswert.
- Die experimentellen Ergebnisse der Ähnlichkeitsmessung sollten in Form von Matrizen der Gestalt  $(s_{ij})_{ij}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq n$ ,  $s_{ij} \in [0, 1]$  weiter zu verarbeiten sein. Daher sind die Eigenschaften eines Ähnlichkeitsmaßes wie z.B.  $s_{ij} = s_{ji}$  und  $s_{ij} \leq s_{jj} = 1$  gefordert.

Bereits im Web Structure Mining verspricht ein solches Verfahren ein hohes Anwendungspotenzial, z.B.:

- Die Bestimmung der strukturellen Ähnlichkeit von web-basierten Dokumentstrukturen wie z.B. Website-Strukturen<sup>1</sup> oder DOM-Trees [45]: Dieser Aufgabe liegt aber das web-basierte Graphmatching zu Grunde, welches in Abbildung (5.1) schematisch dargestellt wird. Ausgehend von einem Testkorpus  $T = \{H_1, H_2, \dots, H_n\}$  und einem unbekannten Eingabegraph  $H_e$  soll nun ein System, in dem ein Ähnlichkeitsmaß  $d$  zwischen zwei Graphen

<sup>1</sup>Damit ist die gesamte Website in Form eines hierarchisierten und gerichteten Graphen gemeint. Siehe Definition (5.3.1) in Kapitel (5.3).

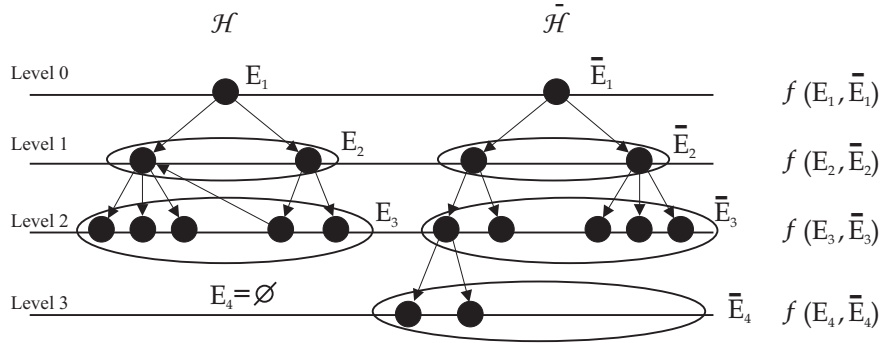
berechnet wird, diejenigen Graphen  $H_i$  bestimmen, für deren Ähnlichkeitswerte mit  $H_e$  die Bedingung  $d(H_e, H_i) \leq g$  oder  $d(H_e, H_i) \geq \bar{g}$ ,  $0 < g, \bar{g} \leq 1$  erfüllt ist.

- Suche und struktureller Vergleich von Graphmustern in web-basierten Hypertextstrukturen: Damit sind auch Interpretationsfragen der Navigationsmuster angesprochen. Einerseits werden in dieser Arbeit mit Hilfe des Verfahrens auf Basis berechneter Ähnlichkeitsmatrizen, die in Kapitel (2.4) dargestellten Clusteringverfahren zur Aufdeckung von strukturell signifikanten Typklassen eingesetzt. Diese Schritte stellen bereits eine deutliche Erweiterung des bekannten Index-Konzepts dar. Andererseits ist in diesem Zusammenhang auch die Aufdeckung und Erforschung graphbasierter Benutzergruppen im Web Usage Mining möglich.
- Besseres Verständnis der graphentheoretischen Struktur von bestehenden Hypertexten: Angenommen, es sei ein Testkorpus  $T$  von graphbasierten Hypertexten gegeben. Dann ist auf der Grundlage des Maßes  $d$  die Bestimmung der Verteilungen der Ähnlichkeitswerte sinnvoll. Damit können wichtige Strukturfragen hinsichtlich der Klassifikation beantwortet werden, z.B.:

*Wieviel Prozent der web-basierten Hypertexte in  $T$  besitzen einen Ähnlichkeitswert kleiner gleich  $\theta \in [0, 1]$ ?*

- Über die Anwendung des Web Structure Mining hinaus kann das dem Verfahren zu Grunde liegende Graphähnlichkeitsmaß auf Grund seiner Konzeption leicht auf neue Graphprobleme in anderen Wissenschaftsbereichen, z.B. bei der Unterscheidung von Tumorstadien [73], angewendet werden. Dabei ist das Graphähnlichkeitsmaß auf baumähnlichen Graphen definiert: Es handelt sich um hierarchisierte und gerichtete Graphen, die Knotenmarkierungen besitzen können. Da die Instanzen dieser Graphklasse sehr häufig in Verbindung graphbasierter Problemstellungen vorkommen, scheint die Gewichtungsmöglichkeit struktureller Aspekte besonders sinnvoll. So können strukturelle Aspekte, die für jede Problemstellung spezifisch sind, optimiert angepasst werden.

Nachdem das Verfahren zur Bestimmung der strukturellen Ähnlichkeit von graphbasierten Dokumentstrukturen aus der Sicht von praktischen Bedingungen und Anwendungen motiviert wurde, wird nun im Folgenden die mathematische Modellierung thematisiert. Um breite Anwendungsfelder zu schaffen, deren graphentheoretische Problemstellungen auf hierarchisierten und gerichteten Graphen beruhen, sollte zu Anfang der Modellierungsphase eine wesentliche Grundeigenschaft berücksichtigt werden. Diese basiert auf einer Folge von zunächst noch



**Abbildung 5.2:** Wichtige Grundeigenschaft des zukünftigen Graphähnlichkeitsmaßes: Die Bildung der  $f(E_i, \bar{E}_i)$  geschieht ebenenweise.

abstrakten Konstruktionsschritten, die durch die Abbildung (5.2) ausgedrückt werden:

- Jeder hierarchisierte und gerichtete Graph wird in Ebenen unterteilt, siehe Abbildung (5.2).
- Auf jeder Ebene  $i$  werden für  $H$  bzw.  $\bar{H}$  spezifische Eigenschaften<sup>2</sup>  $E_i$  bzw.  $\bar{E}_i$  abgeleitet und mittels einer Funktion  $f$  ein Ähnlichkeitswert  $f(E_i, \bar{E}_i) \in [0, 1]$  bestimmt.  $f(E_i, \bar{E}_i)$  bezeichnet dabei den Ähnlichkeitswert auf der  $i$ -ten Ebene, basierend auf  $E_i, \bar{E}_i$ . Ein auszeichnendes Merkmal ist nun: Die Gesamtheit der Werte  $f(E_i, \bar{E}_i)$  bildet nicht *per se* das Maß  $d$ , sondern auf der Grundlage der  $f(E_i, \bar{E}_i)$  kann jederzeit ein neues  $d$  definiert werden.
- Somit ist die Möglichkeit gegeben, für jedes spezielle Graphähnlichkeitsproblem gewisse strukturelle Eigenschaften anders zu berücksichtigen.

Der wesentliche Vorteil einer solchen Konstruktion ist, dass das eigentliche Ähnlichkeitsmaß  $d$  nach speziellen, strukturellen Gesichtspunkten aus den Werten  $f(E_i, \bar{E}_i)$  konstruiert werden kann. Somit ist grundsätzlich die Möglichkeit gegeben, je nach Problemstellung, das Ähnlichkeitsmaß  $d$  so zu konstruieren, dass es gewisse Grapheneigenschaften „schwächer“ oder „stärker“ berücksichtigt.

Auf der Suche nach geeigneten strukturellen Ausprägungen der betrachteten Graphen stellt sich die wichtige und grundlegende Frage:

*Welche Kennzahlen solcher Graphen sind effizient zu berechnen und erlauben die Definition von Ähnlichkeitsmaßen mit möglichst wenig Strukturverlust?*

<sup>2</sup>Diese werden speziell in Kapitel (5.4) definiert.

Betrachtet man zunächst einfache Kennzahlen von Graphen, die etwas über die Graphstruktur aussagen, so sind dies beispielsweise:

- Durchmesser eines Graphen: Es sei  $H = (V, E)$ ,  $E \subseteq V \times V$  ein gerichteter Graph und  $\text{dist}(v, \tilde{v})$ ,  $v, \tilde{v} \in V$  bezeichnet die Länge des minimalen Weges<sup>3</sup> zwischen den Knoten  $v_i$  und  $v_j$ . Falls nun die Zahl  $d(V_1, V_2) := \max\{\text{dist}(v, \tilde{v}) \mid v \in V_1, \tilde{v} \in V_2\}$  die Distanz der beliebigen Knotenmengen  $V_1$  und  $V_2$  definiert, bezeichnet  $\text{diam}(H) := d(V, V)$  den Durchmesser von  $H$ .
- $n$ -Sphäre [103] um den Knoten  $v \in V$  bezogen auf einen Graph  $H$ : Sie ist definiert als die Menge  $D_n(v, H) := \{\tilde{v} \in V \mid \text{dist}(v, \tilde{v}) = n, n \geq 1\}$ . Das heißt, sie umfasst gerade solche Knoten, die mit  $v \in V$  einen minimalen Weg der Länge  $n$  gemeinsam haben.
- Die Höhe eines hierarchisierten Graphen: Sie ist definiert als die Länge des maximalen Weges von der Wurzel zu einem *Blatt*.
- Die Ordnungen der Graphen und spezifische Kantenschnittmengen<sup>4</sup>.

Man sieht sofort, dass sich diese Größen für ganzheitliche Vergleiche solcher Strukturen nicht eignen: Denkbar wäre zwar die Bildung von Gruppen  $G_i$ , die beispielsweise Graphen mit einem bestimmten Durchmesser  $\text{diam}(\mathcal{H})$  oder einer Höhe  $h$  enthalten, um so Graphen mit ähnlichen strukturellen Eigenschaften zu gewinnen. Um aber Graphen ganzheitlich zu vergleichen, erfassen die obigen Größen zu wenig der gemeinsamen Graphstruktur. Im Hinblick auf das zu entwickelnde Ähnlichkeitsmaß, welches nach der Grundidee der Abbildung (5.2) konstruiert wird, sind nun geeignete strukturelle Kenngrößen gesucht. Dazu werden in Kapitel (5.2) die Gradsequenzen<sup>5</sup> von Graphen sowie deren Eigenschaften und Fragestellungen in diesem Problemkreis detailliert betrachtet.

## 5.2 Gradsequenzen von Graphen

Das Konzept sogenannter Gradsequenzen kommt in vielen graphentheoretischen Problemstellungen vor. Sie finden beispielsweise Anwendung bei CHEN [49] zur Aufzählung von chemischen Isomeren [51]. Da in diesem Kapitel hauptsächlich

---

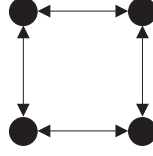
<sup>3</sup>Falls dieser existiert, ansonsten gilt  $\text{dist}(v, \tilde{v}) = \infty$ .

<sup>4</sup>Ein derartiges Ähnlichkeitsmaß, welches auf der Kantenschnittmenge zweier Graphen definiert ist, wurde z.B. von WINNE et al. [247] angegeben. Die Kritikpunkte dieses Maßes wurden bereits in Kapitel (2.3.3), (2.3.4) diskutiert.

<sup>5</sup>Weil hier gerichtete Graphen betrachtet werden, sind damit Sequenzen der Ausgangs- und der Eingangsgrade der Knoten des Graphen gemeint. Siehe Definition (5.2.2) in Kapitel (5.2).



$\mathcal{H}$



**Abbildung 5.3:** Ein gerichteter Graph mit der Ordnung  $|V| = 4$ .

gerichtete Graphen betrachtet werden, sind im Folgenden die meisten Definitionen für gerichtete Graphen formuliert. Um den Begriff der Gradsequenzen formal zu erfassen, benötigt man einige Definitionen wie folgt.

**Definition 5.2.1** Es sei  $\mathcal{H} = (V, E)$ ,  $E \subseteq V \times V$ ,  $|V| < \infty$  ein gerichteter Graph.

$$\begin{aligned}\mathcal{N}^+(v) &:= \{\tilde{v} \in V \setminus \{v\} \mid (v, \tilde{v}) \in E\} \text{ ist die Menge der out-Nachbarn von } v, \\ \mathcal{N}^-(v) &:= \{\tilde{u} \in V \setminus \{v\} \mid (\tilde{u}, v) \in E\} \text{ ist die Menge der in-Nachbarn von } v, \\ \delta_{out}(v) &:= |\mathcal{N}^+(v)|, \\ \delta_{in}(v) &:= |\mathcal{N}^-(v)|.\end{aligned}$$

**Definition 5.2.2** Es sei  $\mathcal{H} = (V, E)$ ,  $E \subseteq V \times V$ ,  $|V| < \infty$  ein gerichteter Graph.  $s_j^{out}(\mathcal{H}) \in \mathbb{N}$ ,  $0 \leq j \leq k_{out} := \max_{v \in V} \{\delta_{out}(v)\}$  bzw.  $s_i^{in}(\mathcal{H}) \in \mathbb{N}$ ,  $0 \leq i \leq k_{in} := \max_{v \in V} \{\delta_{in}(v)\}$  bezeichnet die Anzahl der Knoten von  $\mathcal{H}$  mit Ausgangsgrad  $j$  bzw. mit Eingangsgrad  $i$ .

$$s^{out}(\mathcal{H}) := (s_0^{out}(\mathcal{H}), s_1^{out}(\mathcal{H}), \dots, s_{k_{out}}^{out}(\mathcal{H}))$$

bzw.

$$s^{in}(\mathcal{H}) := (s_0^{in}(\mathcal{H}), s_1^{in}(\mathcal{H}), \dots, s_{k_{in}}^{in}(\mathcal{H}))$$

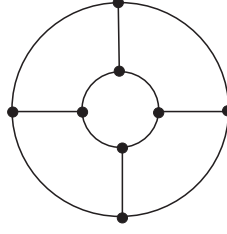
bezeichnet die Ausgangsgrad- bzw. Eingangsgradsequenz von  $\mathcal{H}$ .

Um die Definition (5.2.2) beispielhaft anzuwenden, sei die Abbildung (5.3) betrachtet. Es gilt hier  $s^{out}(\mathcal{H}) = (0, 0, 4) = s^{in}(\mathcal{H}) = (0, 0, 4)$ .

Eine grundlegende und sofort ersichtliche Aussage [186, 187] bezüglich der Gradsequenzen ist

**Proposition 5.2.3** Es sei  $\mathcal{H} = (V, E)$ ,  $E \subseteq V \times V$ ,  $|V| = n$  ein endlicher, gerichteter Graph. Dann gilt

$$\sum_{i=1}^n \delta_{out}(v_i) = \sum_{i=1}^n \delta_{in}(v_i).$$



**Abbildung 5.4:** Ein 3-regulärer Graph.

Die Aussage von Proposition (5.2.3) gilt, da jeder Knoten  $v_i \in V$ , der eine ausgehende Kante besitzt, in  $v_j \in V$  einen Eingangsgrad induziert. Da bei ungerichteten Graphen die Gleichung  $\delta_{out}(v_i) = \delta_{in}(v_i), v_i \in V$  besteht, spricht man hier nur vom *Grad*  $\delta(v_i)$  eines Knotens  $v_i$ . Eine Klasse von ungerichteten Graphen, die durch die Knotengrade charakterisiert werden, sind z.B. die *regulären* Graphen. Ein ungerichteter Graph heißt deshalb *k-regulär*, falls alle Knoten  $v_i \in V, 1 \leq i \leq n$  den Grad  $k$  besitzen. Die Abbildung (5.4) zeigt einen 3-regulären Graph, wobei dieser in der Graphentheorie auch oft als *kubischer Graph* bezeichnet wird.

Die Untersuchung der Gradsequenzen von Graphen tritt oft bei strukturellen Fragestellungen auf, aber auch im Hinblick auf die Realisierbarkeit einer Sequenz natürlicher Zahlen. Für den gerichteten Fall heißt das genauer formuliert: Es seien die Sequenzen natürlicher Zahlen  $s^{(1)} := (s_0, s_1, \dots, s_p)$  und  $s^{(2)} := (\hat{s}_0, \hat{s}_1, \dots, \hat{s}_{\hat{p}})$  gegeben.

*Unter welchen Bedingungen stellen  $s^{(1)}$  und  $s^{(2)}$  Ausgangsgrad- und Eingangsgradsequenzen eines Graphen  $G$  dar?*

Diese Fragestellung wurde beispielsweise für gerichtete und ungerichtete Graphen in [101, 102, 185] untersucht. Für gerichtete Graphen

$$G = (V, E), \quad E \subseteq V \times V, \quad |V| = n, \quad (5.1)$$

untersucht HAKIMI in [102] die Frage der Realisierbarkeit, indem er jedem Knoten  $v_i \in V$  das Paar  $(\delta_{out}(v_i), \delta_{in}(v_i))$  zuordnet. Das bedeutet, dass jeder Knoten durch die Anzahl der ein- und ausgehenden Kanten identifiziert wird. Es sei nun eine Sequenz von 2-Tupeln der Form  $S = (\delta_{out}(v_i), \delta_{in}(v_i)), 1 \leq i \leq n$  nach der wachsenden Komponentensumme geordnet. Dann ist ein Hauptergebnis dieser Arbeit, dass die Sequenz  $S$  durch einen schlingenfreien Graph (5.1) realisierbar ist, genau dann, wenn die Gleichung

$$\sum_{i=1}^n \delta_{out}(v_i) = \sum_{i=1}^n \delta_{in}(v_i), \quad (5.2)$$

und die Ungleichung

$$\sum_{i=1}^{n-1} \delta_{out}(v_i) + \delta_{in}(v_i) \geq \delta_{out}(v_n) + \delta_{in}(v_n) \quad (5.3)$$

gilt. Weitergehend weist HAKIMI in [102] nach, dass ein zusammenhängender gerichteter Graph realisierbar ist, falls die Gleichung (5.2), die Ungleichung (5.3) und zusätzlich die Ungleichung

$$\sum_{i=1}^n \delta_{out}(v_i) \geq n - 1 \quad (5.4)$$

erfüllt ist. Ist  $S$  durch einen streng zusammenhängenden Graph realisierbar, dann gilt Gleichung (5.2), die Ungleichung (5.3) und

$$\min(\delta_{out}(v_i), \delta_{in}(v_i)) > 0 \quad \forall i : 1 \leq i \leq n. \quad (5.5)$$

Ein Ergebnis für ungerichtete Graphen, das als letztes in diesem Problemkreis genannt werden soll, ist ein Satz von ERDÖS und DALLAI [239]. Der Beweis ist in [239] zu finden.

**Satz 5.2.4** *Eine Sequenz  $\delta(v_1) \geq \delta(v_2) \geq \dots \geq \delta(v_n)$  ganzer natürlicher Zahlen realisiert einen Graphen ohne Schlingen und Mehrfachkanten genau dann, wenn  $\sum_{i=1}^n \delta(v_i)$  gerade ist und wenn die Ungleichung*

$$\sum_{i=1}^n \delta(v_i) \leq p(p-1) + \sum_{i=p+1}^n \min\{p, \delta(v_i)\}, \quad \forall p : 1 \leq p \leq n$$

*gilt.*

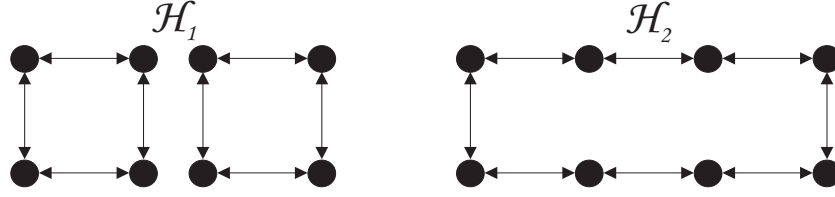
Eine für diese Arbeit grundlegende Fragestellung ist, wie Aussagen über Gradsequenzen mit denen der Graphisomorphie wechselwirken. Von besonderem Interesse sind dabei solche Aussagen, aus denen auf der Basis von gewissen Bedingungen der Gradsequenzen Schlüsse bezüglich der Graphisomorphie gezogen werden können. Abschließend wird aus diesem Problemkreis die folgende Proposition angegeben:

**Proposition 5.2.5** *Es seien  $\mathcal{H}_1$  und  $\mathcal{H}_2$  endliche und gerichtete Graphen. Falls  $\phi$  der Isomorphismus von  $\mathcal{H}_1$  auf  $\mathcal{H}_2$  ist, so gilt für  $v_i, 1 \leq i \leq n$*

$$\delta_{out}(v_i) = \delta_{out}(\phi(v_i)) \quad (5.6)$$

*und*

$$\delta_{in}(v_i) = \delta_{in}(\phi(v_i)). \quad (5.7)$$



**Abbildung 5.5:** Zwei nicht isomorphe Graphen mit den gleichen Ausgangsgrad- und Eingangsgradsequenzen.

Aus Proposition (5.2.5) und mit der Definition der Isomorphie folgt nun unmittelbar die Aussage:  $\mathcal{H}_1 \cong \mathcal{H}_2 \implies s^{out}(\mathcal{H}_1) = s^{out}(\mathcal{H}_2) \wedge s^{in}(\mathcal{H}_1) = s^{in}(\mathcal{H}_2)$ . Das heißt, dass aus der Isomorphie von  $\mathcal{H}_1$  und  $\mathcal{H}_2$  folgt, dass diese Graphen gleiche Ausgangsgrad- und Eingangsgradsequenzen besitzen. Der einfache Beweis der Proposition ergibt sich aus der Isomorphiebedingung von  $\mathcal{H}_1$  und  $\mathcal{H}_2$ . Die Umkehrung der Proposition gilt jedoch nicht notwendigerweise, wie Abbildung (5.5) zeigt: Für die Ausgangsgrad- und Eingangsgradsequenzen erhält man  $s^{out}(\mathcal{H}_1) = (0, 0, 8) = s^{out}(\mathcal{H}_2) = (0, 0, 8)$ ,  $s^{in}(\mathcal{H}_1) = (0, 0, 8) = s^{in}(\mathcal{H}_2) = (0, 0, 8)$ . Da  $\mathcal{H}_1$  nicht zusammenhängend ist, gilt offensichtlich  $\mathcal{H}_1 \not\cong \mathcal{H}_2$ .

### 5.3 Hierarchisierte und gerichtete Graphen

Das Beispiel aus Abbildung (5.5) wirft unmittelbar die Frage auf, „wieviel Struktur“ die Ausgangsgrad- und Eingangsgradsequenzen eines Graphen erfassen. Um vertiefende Beispiele anzugeben, wird im Folgenden die Graphklasse der knotenmarkierten, hierarchisierten und gerichteten Graphen formal definiert. In Kapitel (3.2) wurden diese Graphen bereits im Zusammenhang mit der web-basierten Extraktion erwähnt.

**Definition 5.3.1** *Es sei die Knotenmenge*

$$\hat{V} := \{v_{0,1}, v_{1,1}, v_{1,2}, \dots, v_{1,\sigma_1}, v_{2,1}, v_{2,2}, \dots, v_{2,\sigma_2}, \dots, v_{h,1}, v_{h,2}, \dots, v_{h,\sigma_h}\},$$

eine Knotenmarkierungsfunktion  $m_{\hat{V}} : \hat{V} \longrightarrow A_{\hat{V}}$  und ein Knotenalphabet  $A_{\hat{V}}$  gegeben.  $h$  bezeichnet die maximale Länge eines Pfades von der Wurzel  $v_{0,1}$  bis zu einem Blatt.  $v_{i,j}$  bezeichnet den  $j$ -ten Knoten auf der  $i$ -ten Ebene,  $0 \leq i \leq h$ ,  $1 \leq j \leq \sigma_i$ .  $\sigma_i$  ist maximal in dem Sinne, dass keine andere Knotensequenz existiert, so dass  $v_{i,1}, v_{i,2}, \dots, v_{i,\hat{\sigma}_i}$  mit  $\hat{\sigma}_i > \sigma_i$ .  $\mathcal{L} : \hat{V} \longrightarrow \mathbb{N}$ ,  $\mathcal{L}(v_{i,j}) := i$  ist eine Funktion, welche die Ebene eines Knotens  $v_{i,j}$  bestimmt. Die Kantenmenge  $\hat{E} := \hat{E}_1 \cup \hat{E}_2 \cup \hat{E}_3 \cup \hat{E}_4$  sei wie folgt definiert:

$$\begin{aligned}\hat{E}_1 := \{ & (v_{i,\nu}, v_{i+1,\nu_j}) | v_{i,\nu}, v_{i+1,\nu_j} \in \hat{V}, 1 \leq j \leq k, k := \delta_{out}(v_{i,\nu}), \\ & \mathcal{L}(v_{i+1,\nu_j}) = \mathcal{L}(v_{i,\nu}) + 1 \wedge ((\bar{\mathcal{A}}(v_{i,\bar{\nu}}, v_{i+1,\nu_k}), \bar{\nu} > \nu) \vee \\ & (\bar{\mathcal{A}}(v_{i,\hat{\nu}}, v_{i+1,\nu_1}), \hat{\nu} < \nu)), \nu_1 < \nu_2 < \dots < \nu_k \} \quad (5.8)\end{aligned}$$

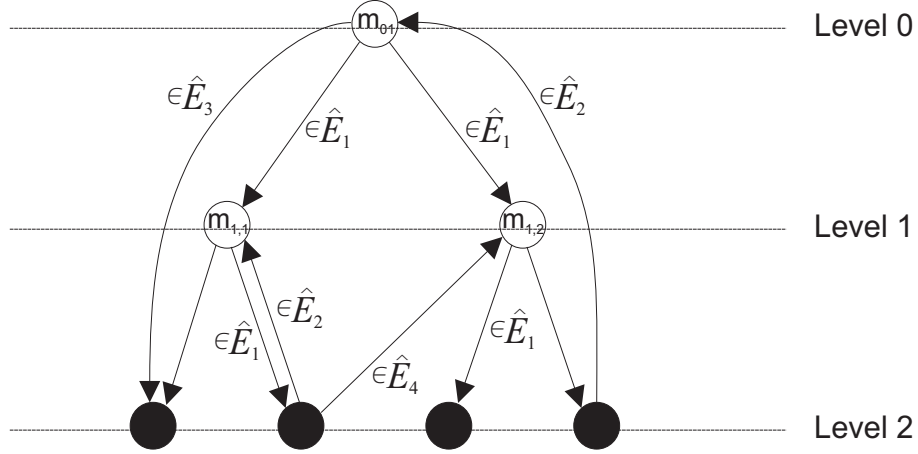
$$\begin{aligned}\hat{E}_2 := \{ & (v_{i+s,\nu}, v_{i,\bar{\nu}}) | v_{i+s,\nu}, v_{i,\bar{\nu}} \in \hat{V}, \mathcal{L}(v_{i,\bar{\nu}}) = \mathcal{L}(v_{i+s,\nu}) - s, s \leq h \\ & \wedge \exists! (\underbrace{(v_{i,\bar{\nu}}, v_{i+1,\nu_1})}_{\in \hat{E}_1}, \dots, \underbrace{(v_{i+s-1,\nu_j}, v_{i+s,\nu})}_{\in \hat{E}_1}), 1 \leq \bar{\nu} \leq \sigma_i, \\ & 1 \leq \nu_1 \leq \sigma_{i+1}, \dots, 1 \leq \nu_j \leq \sigma_{i+s-1}, 1 \leq \nu \leq \sigma_{i+s} \}. \quad (5.9)\end{aligned}$$

$$\begin{aligned}\hat{E}_3 := \{ & (v_{i,\bar{\nu}}, v_{i+s,\nu}) | v_{i,\bar{\nu}}, v_{i+s,\nu} \in \hat{V}, \mathcal{L}(v_{i+s,\nu}) = \mathcal{L}(v_{i,\bar{\nu}}) + s, 1 < s \leq h \\ & \wedge \exists! (\underbrace{(v_{i,\bar{\nu}}, v_{i+1,\nu_1})}_{\in \hat{E}_1}, \dots, \underbrace{(v_{i+s-1,\nu_j}, v_{i+s,\nu})}_{\in \hat{E}_1}), 1 \leq \bar{\nu} \leq \sigma_i, \\ & 1 \leq \nu_1 \leq \sigma_{i+1}, \dots, 1 \leq \nu_j \leq \sigma_{i+s-1}, 1 \leq \nu \leq \sigma_{i+s} \}. \quad (5.10)\end{aligned}$$

$$\begin{aligned}\hat{E}_4 := \{ & (v_{i,\nu}, v_{i,\bar{\nu}}) | v_{i,\nu}, v_{i,\bar{\nu}} \in \hat{V}, \mathcal{L}(v_{i,\nu}) = \mathcal{L}(v_{i,\bar{\nu}}) \wedge (\nu < \bar{\nu} \vee \nu > \bar{\nu}) \} \\ & \cup \{ (v_{i+s,\nu}, v_{i,\bar{\nu}}) | v_{i+s,\nu}, v_{i,\bar{\nu}} \in \hat{V}, (v_{i+s,\nu}, v_{i,\bar{\nu}}) \notin \hat{E}_2, \\ & \mathcal{L}(v_{i,\bar{\nu}}) = \mathcal{L}(v_{i+s,\nu}) - s, s \leq h \} \\ & \cup \{ (v_{i,\nu}, v_{i+s,\bar{\nu}}) | v_{i,\nu}, v_{i+s,\bar{\nu}} \in \hat{V}, (v_{i,\nu}, v_{i+s,\bar{\nu}}) \notin \hat{E}_1, \hat{E}_3, \\ & \mathcal{L}(v_{i+s,\bar{\nu}}) = \mathcal{L}(v_{i,\nu}) + s, s \leq h \}. \quad (5.11)\end{aligned}$$

Dann bezeichnet  $\hat{\mathcal{H}}_m = (\hat{V}, \hat{E}, m_{\hat{V}}, A_{\hat{V}})$  den knotenmarkierten, hierarchisierten und gerichteten Graph. Falls  $\hat{\mathcal{H}}_m$  ohne Knotenmarkierung aufgefasst wird, gilt  $A_{\hat{V}} := \{\}$ .

Die Abbildung (5.6) zeigt beispielhaft einen knotenmarkierten, hierarchisierten und gerichteten Graph, zusammen mit seinen Kantentypen. Eine informelle Erklärung der Kantentypen aus Definition (5.3.1) kann wie folgt formuliert werden [156]:



**Abbildung 5.6:** Kantentypen eines Graphen nach Definition (5.3.1).

- $(\hat{E}_1)$  *Kernel-Kanten*: Die so genannte *Kernel-Hierarchie* wird durch die *Kernel-Kanten* aufgespannt. Die Kernel-Hierarchie entspricht der Graphstruktur eines gerichteten Wurzelbaums. *Kernel-Kanten* verbinden die Knoten der Kernel-Hierarchie mit ihrem unmittelbaren Nachfolgerknoten<sup>6</sup>.
- $(\hat{E}_2)$  *Up-Kanten* verbinden sinngemäß Knoten, denen in umgekehrter Richtung eine Folge von Kernel-Kanten zu Grunde liegt. Mit anderen Worten: Sie verbinden Knoten der Kernel-Hierarchie mit einem Vorgängerknoten der Kernel-Hierarchie.
- $(\hat{E}_3)$  *Down-Kanten* verbinden Knoten der Kernel-Hierarchie mit einem Nachfolgerknoten der Kernel-Hierarchie. Ihnen liegt in richtiger Richtung eine Folge von Kernel-Kanten zu Grunde.
- $(\hat{E}_4)$  *Across-Kanten* verbinden Knoten der Kernel-Hierarchie, wobei kein Knoten ein unmittelbarer Vorgänger in Bezug auf die zu Grunde liegende Kernel-Hierarchie ist.

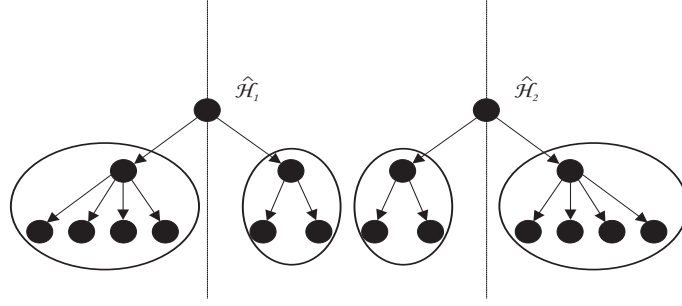
Da es sich bei den Graphen der Definition (5.3.1) um baumähnliche Strukturen handelt, liegt die folgende Aussage auf der Hand.

**Proposition 5.3.2** *Es sei  $\hat{\mathcal{H}}_m = (\hat{V}, \hat{E}, m_{\hat{V}}, A_{\hat{V}})$ . Dann ist*

$$\hat{\mathcal{H}}_m^T := (\hat{V}, E_T, m_{\hat{V}}, A_{\hat{V}}), \quad E_T := \hat{E} \setminus \{\hat{E}_2, \hat{E}_3, \hat{E}_4\}$$

*ein gerichteter Wurzelbaum, zyklensfrei und es gilt  $|E_T| = |\hat{V}| - 1$ .*

<sup>6</sup>Im Sinne der Kernel-Hierarchie.



**Abbildung 5.7:** Zwei nicht symmetrische Graphen mit den gleichen Ausgangsgrad- und Eingangsgradsequenzen.

Daraus folgt ebenfalls unmittelbar eine rekursive Definition der Wurzelbaumstruktur.

**Corollar 5.3.3** *Es sei  $w := v_{0,1}$ . Dann ist*

$$\hat{\mathcal{H}}_m^{T_{rec}} := \left( w, \hat{\mathcal{H}}_m^{T_1}, \hat{\mathcal{H}}_m^{T_2}, \dots, \hat{\mathcal{H}}_m^{T_{\delta_{out}(w)}} \right)$$

*eine rekursive Definition von  $\hat{\mathcal{H}}_m^T$ .*

Im Folgenden wird beispielhaft die Fragestellung wieder aufgegriffen, wie stark die Beziehungen zwischen Ausgangs- und Eingangsgradsequenzen auf die Graph-topologie einwirken. Die Graphen aus Abbildung (5.7) besitzen die gleichen Ausgangsgrad- und Eingangsgradsequenzen, es gilt:

$$\begin{aligned} s^{out}(\hat{\mathcal{H}}_1) &= (6, 0, 2, 0, 1) = s^{out}(\hat{\mathcal{H}}_2) = (6, 0, 2, 0, 1) \wedge \\ s^{in}(\hat{\mathcal{H}}_1) &= (1, 8) = s^{in}(\hat{\mathcal{H}}_2) = (1, 8). \end{aligned}$$

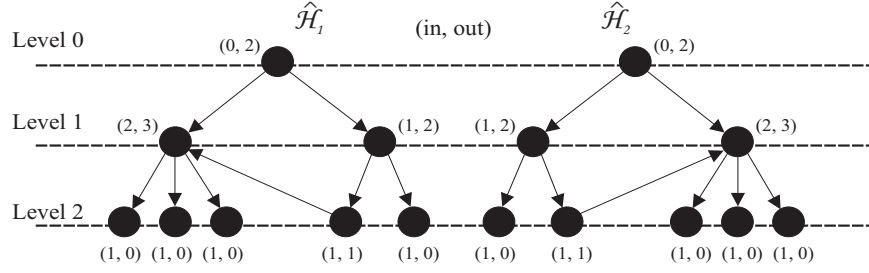
Da in einem Gradsequenzvektor<sup>7</sup> eines gerichteten Graphen  $\mathcal{H}$  nur die Anzahl  $s_j^{out}(\mathcal{H})$  der Knoten mit Ausgangsgrad  $j$  bzw.  $s_i^{in}(\mathcal{H})$  mit Eingangsgrad  $i$  gezählt werden, wird die Nichtsymmetrie<sup>8</sup> durch die Gleichheit der Ausgangsgrad- und Eingangsgradsequenzen nicht erfasst. Damit sind einfache Vergleiche von Gradsequenzvektoren zur Durchführung von Graphvergleichen unzureichend.

## 5.4 Zentraler Lösungsansatz

In Kapitel (5.1) wurde im Hinblick auf die Konstruktion der neuen Methode zur Bestimmung der Ähnlichkeit strukturierter Objekte eine abstrakte Idee dargestellt, die im Wesentlichen aus zwei Schritten besteht:

<sup>7</sup>Siehe Definition (5.2.2) in Kapitel (5.2).

<sup>8</sup>Die Symmetrieachsen der Graphen aus Abbildung (5.7) sind durch vertikale Linien angedeutet.



**Abbildung 5.8:** Induzierte Ausgangsgrad- und Eingangsgradsequenzen auf den Ebenen.

1. Die Definition einer Transformation  $T : O \longrightarrow N$ , die Objekte  $O_i \in O$  in einen niedrigdimensionaleren Objektraum  $N$  abbildet.
2. Gesucht sind nun bekannte oder neue Methoden, um die Ähnlichkeit der Objekte  $N_i \in N$  jetzt effizienter und mit möglichst wenig Strukturverlust zu berechnen.

Weiterhin wurde in Kapitel (4.4) diskutiert, dass Ähnlichkeitsmaße, die auf Graphisomorphie beruhen, auf Grund ihres ungenügenden Komplexitätsverhaltens für das web-basierte Graphmatching ungeeignet sind. Der obigen Konstruktionsidee folgend muss nun eine geeignete Transformation gewählt werden, um hierarchisierte und gerichtete Graphen in eindimensionale Strukturen – hier werden *formale Zeichenketten* gewählt – abzubilden. Bevor mit der Darstellung des zentralen Lösungsansatzes begonnen wird, folgt zunächst eine Definition, um die Begriffe Sequenz und Alignment besser zu erfassen.

**Definition 5.4.1 (Sequenz und Sequenz-Alignment)** *Unter einer Sequenz versteht man im Folgenden ein Wort über einem beliebig gewählten Alphabet. Eine Zuordnung von Entsprechungen zwischen den Bausteinen von Sequenzen wird als Sequenz-Alignment, oder falls kein Konflikt besteht, als Alignment bezeichnet.*

Nun betrachte man beispielhaft die hierarchisierten und gerichteten Graphen aus Abbildung (5.8). Unterteilt man nun die Graphen ebenenweise, so lässt sich die  $i$ -te Ebene als formale Knotensequenz<sup>9</sup>  $v_{i,1}, v_{i,2}, \dots, v_{i,\sigma_i}$  beschreiben. Das impliziert aber, dass jede Knotensequenz der Form  $v_{i,1}, v_{i,2}, \dots, v_{i,\sigma_i}$ , Ausgangsgrad- und Eingangsgradsequenzen auf der Ebene  $i, 0 \leq i \leq h$  induzieren. Die Abbildung (5.8) zeigt die durch die entsprechenden Knotensequenzen induzierten Ausgangs- und Eingangsgrade in Tupelform.

<sup>9</sup>Das sind ebenfalls Wörter über einem speziell gewählten Alphabet. Im Hinblick auf das Alignment von induzierten Ausgangsgrad- und Eingangsgradsequenzen auf der Ebene  $i$  sei angemerkt, dass diese Gradsequenzen unabhängig von evtl. vorhandenen Knotenmarkierungen existieren.



Wendet man diese Betrachtungsweise auf die gesamte Graphstruktur an, so folgt für beliebige Graphen  $\hat{\mathcal{H}}_m^1$  und  $\hat{\mathcal{H}}_m^2$ , die der Definition (5.3.1) genügen, die Darstellung:

$$\begin{aligned} S_0^{\hat{\mathcal{H}}_m^1} &:= w_1^{\hat{\mathcal{H}}_m^1}, \\ S_1^{\hat{\mathcal{H}}_m^1} &:= v_{1,1}^{\hat{\mathcal{H}}_m^1} \circ v_{1,2}^{\hat{\mathcal{H}}_m^1} \circ \dots \circ v_{1,\delta_{out}(w_1^{\hat{\mathcal{H}}_m^1})}^{\hat{\mathcal{H}}_m^1}, \\ &\vdots \\ S_{h_1}^{\hat{\mathcal{H}}_m^1} &:= v_{h_1,1}^{\hat{\mathcal{H}}_m^1} \circ v_{h_1,2}^{\hat{\mathcal{H}}_m^1} \circ \dots \circ v_{h_1,\sigma_{h_1}}^{\hat{\mathcal{H}}_m^1} \end{aligned}$$

und

$$\begin{aligned} S_0^{\hat{\mathcal{H}}_m^2} &:= w_2^{\hat{\mathcal{H}}_m^2}, \\ S_1^{\hat{\mathcal{H}}_m^2} &:= v_{1,1}^{\hat{\mathcal{H}}_m^2} \circ v_{1,2}^{\hat{\mathcal{H}}_m^2} \circ \dots \circ v_{1,\delta_{out}(w_2^{\hat{\mathcal{H}}_m^2})}^{\hat{\mathcal{H}}_m^2}, \\ &\vdots \\ S_{h_2}^{\hat{\mathcal{H}}_m^2} &:= v_{h_2,1}^{\hat{\mathcal{H}}_m^2} \circ v_{h_2,2}^{\hat{\mathcal{H}}_m^2} \circ \dots \circ v_{h_2,\sigma_{h_2}}^{\hat{\mathcal{H}}_m^2}. \end{aligned}$$

Dabei gelte die Definition  $w_k^{\hat{\mathcal{H}}_m^k} := v_{0,1}^{\hat{\mathcal{H}}_m^k}$ ,  $k \in \{1, 2\}$  und es sei  $v_{i,j}^{\hat{\mathcal{H}}_m^1}$ ,  $0 \leq i \leq h_1$ ,  $1 \leq j \leq \sigma_i$ <sup>10</sup> der  $j$ -te Knoten auf der  $i$ -ten Ebene von  $\hat{\mathcal{H}}_m^1$ . Diese Definition gilt für  $v_{i,j}^{\hat{\mathcal{H}}_m^2}$  aus  $\hat{\mathcal{H}}_m^2$  analog. Da somit eine gewünschte Transformation eines hierarchisierten und gerichteten Graphen in eine Folge von formalen Knotensequenzen durchgeführt wurde, ist nun ein geeignetes Verfahren zu wählen, um die Ähnlichkeit der transformierten Zeichenketten zu bestimmen. Diese Aufgabe wird im Folgenden durch Sequenz-Alignments [145] realisiert. Um nun die strukturelle Ähnlichkeit der Ursprungsgraphen  $\hat{\mathcal{H}}_m^1$  und  $\hat{\mathcal{H}}_m^2$  zu bestimmen, ist damit ein optimales<sup>11</sup> Alignment der formalen Zeichenketten bezüglich einer Kostenfunktion  $\alpha$ , gesucht. Das heißt aber: Je kostengünstiger<sup>12</sup> die Sequenz-Alignments der induzierten Ausgangsgrad- und Eingangsgradsequenzen auf Ebene  $i$ ,  $0 \leq i \leq \rho := \max(h_1, h_2)$  sind, desto ähnlicher ist die gemeinsame Struktur von  $\hat{\mathcal{H}}_m^1$  und  $\hat{\mathcal{H}}_m^2$ . Um solche Alignments durchzuführen, müssen Funktionen<sup>13</sup> zur Bewertung der Alignments definiert werden. Somit kann insbesondere auf Ebene  $i$  ein Ähnlichkeitswert der Alignments bestimmt werden. Damit ist aber die erwünschte Grundeigenschaft, die durch Abbildung (5.2) ausgedrückt wird, erfüllt: Die Werte, die die Güte der Alignments der Ausgangsgrad- und Eingangsgradsequenzen auf den Ebenen detektieren, bilden nicht automatisch das angestrebte Graphähnlichkeitsmaß  $d$ .

<sup>10</sup> $\sigma_i$  bezeichnet wieder den maximalen Index auf der Ebene  $i$ .

<sup>11</sup>Das heißt unter minimalen Kosten. Siehe dazu Kapitel (5.5).

<sup>12</sup>Im Hinblick auf Ähnlichkeitsfunktionen, die die Güte solcher Alignments auswerten. Siehe Kapitel (5.6).

<sup>13</sup>Siehe Kapitel (5.6).

Die Problemstellung, die Bestimmung des optimalen Alignments zwischen zwei Graphen auf der Grundlage der entsprechenden Wort-Repräsentationen als Optimierungsproblem aufzufassen, lautet nun:

*Ausgehend von einem zu Grunde liegenden Sequenz-Alignment ist derjenige Pfad im Alignment-Graph<sup>14</sup> gesucht, der mit minimalen Kosten bewertet wird.*

Zur Lösung des Optimierungsproblems wird das Verfahren der dynamischen Programmierung [15] eingesetzt, wobei die dynamische Programmierung aus der Klasse der *bottom-up-Algorithmen* stammt.

## 5.5 Berechnungsgrundlagen

Im vorliegenden Kapitel (5.5) wird der Grundstein der dynamischen Programmierung gelegt und ein bekannter Algorithmus zur Berechnung optimaler Sequenz-Alignments eingeführt. Probleme bezüglich optimaler Sequenz-Alignments wurden in der Fachliteratur intensiv untersucht [99, 190, 191]. In der vorliegenden Arbeit werden Sequenz-Alignments zur Lösung einer neuen und aktuellen Problemstellung verwendet: Die Bestimmung der strukturellen Ähnlichkeit hierarchisierter und gerichteter Graphen. Dazu werden im ersten Schritt die Graphen in formale Knotensequenzen abgebildet. Im zweiten Schritt wird auf der Basis von Sequenz-Alignments der induzierten Ausgangsgrad- und Eingangsgradsequenzen auf den Ebenen die strukturelle Ähnlichkeit der Graphen bestimmt. Da die Berechnung solcher Sequenz-Alignments effizient ist, kann somit im Gegensatz zu bekannten Methoden die auf Isomorphiebeziehungen beruhen, eine drastische Reduktion der Berechnungskomplexität erreicht werden. Weiterhin besitzt ein effizientes und aussagekräftiges Verfahren zur Bestimmung der strukturellen Ähnlichkeit hierarchisierter und gerichteter Graphen ein hohes Anwendungspotenzial, da graphorientierte Problemstellungen dieser Graphklasse in vielen wissenschaftlichen Bereichen vorkommen.

Zunächst wird die dynamische Programmierung (DP) informell und *optimierungstheoretisch* motiviert, wobei das Hauptziel die Entwicklung algorithmischer Vorschriften zur Berechnung optimaler Sequenz-Alignments ist. Die dynamische Programmierung, die als Optimierungsverfahren bezeichnet werden kann, besitzt in den unterschiedlichsten Wissenschaftsbereichen viele Anwendungen [163], z.B. in der Betriebswirtschaft, in der Biologie, in der Informatik und in den Ingenieurwissenschaften. Dabei legt BELLMAN 1957 den Grundstein der dynamischen

---

<sup>14</sup>Siehe Kapitel (5.5).

Programmierung, indem er das für viele Optimierungsverfahren wichtige BELLMAN'sche Optimalitätsprinzip [15, 16, 163] formuliert. Ausgehend von diskreten, deterministischen und mehrstufigen Entscheidungsprozessen,<sup>15</sup> drückt BELLMAN die Grundeigenschaften optimaler *Entscheidungspolitiken*<sup>16</sup> folgendermaßen aus [15, 16]:

*„Eine optimale Entscheidungspolitik hat die Eigenschaft, dass ungeachtet des Anfangszustandes und der ersten Entscheidung, die verbleibenden Entscheidungen eine optimale Entscheidungspolitik hinsichtlich des aus der ersten Entscheidung resultierenden Zustandes darstellen.“*

Der Beweis, die Formalisierung und die mathematischen Zusammenhänge des Prinzips werden in [15, 16] dargestellt. Mit anderen Worten sagt das eben formulierte Optimalitätsprinzip aus: Teillösungen von Optimallösungen sind selbst optimal und Optimallösungen setzen sich sukzessiv aus Teillösungen zusammen. Diese Aussage und die Additivitätsforderung der Zielfunktion macht sich die dynamische Programmierung zu Nutze. Die dynamische Programmierung gehört dabei zur Gruppe der bottom-up-Verfahren. Das bedeutet, dass für das betrachtete Optimierungsproblem zunächst alle relevanten Teilprobleme gelöst und diese in Tabellenform gespeichert werden. Nach SCHÖNING [199] zerlegt sich die Designphase eines Algorithmus, der auf dynamischer Programmierung beruht, in folgende Schritte:

- Charakterisierung des Lösungsraums und Struktur der gesuchten optimalen Lösung.
- Rekursive Definition der optimalen Lösung. Sie setzt sich rekursiv aus kleineren Optimallösungen zusammen.
- Konzeption des Algorithmus in bottom-up-Form, so dass  $n$  optimale Teillösungen  $T_1, T_2, \dots, T_n$  zusammen mit ihren Werten in Tabellenform gespeichert werden. Wird nun die optimale Teillösung  $T_k, k > 1$  gesucht, nutzt die dynamische Programmierung das Prinzip aus, dass die optimalen Teillösungen  $T_1, T_2, \dots, T_{k-1}$  bereits berechnet wurden.

In der Informatik gibt es viele Algorithmen, die auf dynamischer Programmierung beruhen, z.B.: Das *Rucksack-Problem*, *optimale binäre Suchbäume* und der COOKE-YOUNGER-KASAMI-Algorithmus (CYK). Großen Bekanntheitsgrad hat die dynamische Programmierung durch ihre Anwendungen in der Biologie und in

---

<sup>15</sup>Ein diskreter, deterministischer und mehrstufiger Entscheidungsprozess ist ein Prozess, in dem endlich oder höchstens abzählbar viele Entscheidungen den Prozessverlauf bestimmen.

<sup>16</sup>So bezeichnet BELLMAN [15, 16] eine Folge von zulässigen Entscheidungen  $(q_1, q_2, \dots, q_N)$ .

der Bioinformatik. Dort wird sie vor allem zur Berechnung biologischer Sequenz-Alignments eingesetzt [2, 162, 190, 191].

Um im Nachfolgenden das optimale Sequenz-Alignment der gegebenen Graphen  $\hat{\mathcal{H}}_m^1$  und  $\hat{\mathcal{H}}_m^2$  zu bestimmen, betrachte man  $\hat{\mathcal{H}}_m^1$  und  $\hat{\mathcal{H}}_m^2$  in der Darstellung

$$S_1 := w_1^{\hat{\mathcal{H}}_m^1} \circ v_{1,1}^{\hat{\mathcal{H}}_m^1} \circ v_{1,2}^{\hat{\mathcal{H}}_m^1} \circ \dots \circ v_{h_1, \sigma_{h_1}}^{\hat{\mathcal{H}}_m^1}, \quad (5.12)$$

$$S_2 := w_2^{\hat{\mathcal{H}}_m^2} \circ v_{1,1}^{\hat{\mathcal{H}}_m^2} \circ v_{1,2}^{\hat{\mathcal{H}}_m^2} \circ \dots \circ v_{h_2, \sigma_{h_2}}^{\hat{\mathcal{H}}_m^2}, \quad (5.13)$$

wobei die Graphen jeweils in formale Knotensequenzen transformiert sind. Auf der Basis der Definitionen von  $S_1$  und  $S_2$  bezeichne  $S_k[i]$  die  $i$ -te Position der Sequenzen  $S_k$  und es gelte  $S_1[n] = v_{h_1, \sigma_{h_1}}^{\hat{\mathcal{H}}_m^1}$ ,  $S_2[m] = v_{h_2, \sigma_{h_2}}^{\hat{\mathcal{H}}_m^2}$ ,  $\mathbb{N} \ni n, m \geq 1$ ,  $S_k[1] = w_k^{\hat{\mathcal{H}}_m^k}$ ,  $k \in \{1, 2\}$ .

In Vorbereitung auf den gesuchten Algorithmus, der das optimale Alignment zwischen den Sequenzen (5.12), (5.13) bestimmt, folgt zunächst die Definition des Alignment-Graphen. Dieser verdeutlicht einfache Zusammenhänge zwischen optimalen Sequenz-Alignments und der auf der Basis einer Kostenfunktion minimal bewerteter Pfade.

**Definition 5.5.1 (Alignment-Graph)** *Es sei  $V_{S_1, S_2} := \{(i, j) | 0 \leq i \leq n, 0 \leq j \leq m\}$ ,  $e_{Del} := (i-1, j) \rightarrow (i, j)$ ,  $e_{Ins} := (i, j-1) \rightarrow (i, j)$ ,  $e_{Subst} := (i-1, j-1) \rightarrow (i, j)$  und  $f_{E_{S_1, S_2}} : E_{S_1, S_2} \rightarrow \mathbb{R}_+$  eine Kantenmarkierungsfunktion. Die Kantenmenge  $E_{S_1, S_2}$  wird vollständig definiert durch*

$$\begin{aligned} E_{S_1, S_2} := & \{e_{Del} | f_{E_{S_1, S_2}}(e_{Del}) = [S_1[i], -], i \in [1, n]\} \\ & \cup \{e_{Ins} | f_{E_{S_1, S_2}}(e_{Ins}) = [-, S_2[j]], j \in [1, m]\} \\ & \cup \{e_{Subst} | f_{E_{S_1, S_2}}(e_{Subst}) = [S_1[i], S_2[j]], i \in [1, n], j \in [1, m]\}. \end{aligned}$$

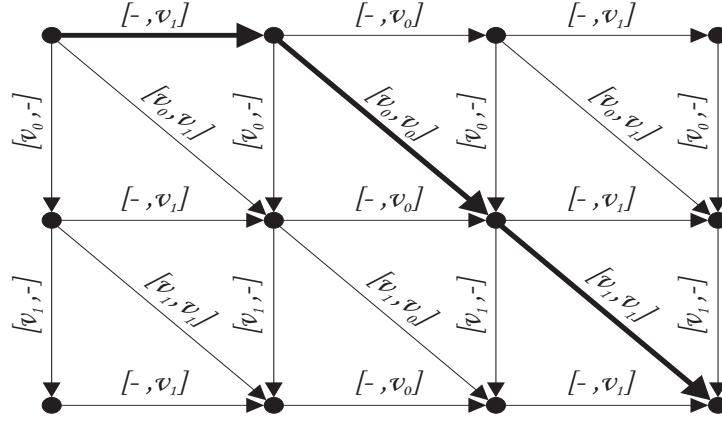
$G_{S_1, S_2} := (V_{S_1, S_2}, E_{S_1, S_2}, f_{E_{S_1, S_2}})$  heißt *Alignment-Graph der Sequenzen  $S_1$  und  $S_2$* .

Die Kanten des Graphen haben dabei operationale Bedeutungen bezüglich  $S_1$  und  $S_2$ . Es gilt:  $(i-1, j) \rightarrow (i, j)$  entspricht der Löschung von  $S_1[i]$  in  $S_1$ ,  $(i, j-1) \rightarrow (i, j)$  entspricht der Einfügung von  $S_2[j]$  in  $S_1$  an der  $i$ -ten Position und  $(i-1, j-1) \rightarrow (i, j)$  entspricht der Ersetzung von  $S_1[i]$  durch  $S_2[j]$ .

Eine zentrale Eigenschaft solcher Alignment-Graphen ist, dass jeder Pfad mit insgesamt minimalen Kosten von der Position  $(0,0)$  zum Zielknoten  $(n, m)$ <sup>17</sup> ein

---

<sup>17</sup>Ein Alignment-Graph spannt eine matrixenähnliche Struktur auf. Den Knoten des Alignment-Graphen kann man anschaulich Matrixpositionen  $(i, j)$  zuordnen. Ein Beispiel zeigt die Abbildung (5.9).



**Abbildung 5.9:** Der Alignment-Graph  $G_{S_1, S_2}$  der Sequenzen  $S_1, S_2$ .

*Edit Transscript* – mit minimaler Zahl von Editieroperationen – darstellt. GUSFIELD [99] definiert das Edit Transscript wie folgt: Es sei  $w_1$  ein Wort über einem gewählten Alphabet und eine Folge von Editieroperationen I, D, R, M gegeben. I (insert) bezeichnet die Einfügung, D (deletion) bezeichnet die Löschung, R (replace) bezeichnet die Ersetzung und M (match) bezeichnet die Übereinstimmung. Die Transformation von  $w_1$  in das Wort  $w_2$  auf der Basis dieser Editieroperationen heißt Edit Transscript. Dabei drückt der folgende Satz von GUSFIELD [99] einen einfachen Zusammenhang zwischen optimalen Sequenz-Alignments und minimal bewerteter Pfade aus.

**Satz 5.5.2** *Es seien die Sequenzen  $S_1$  bzw.  $S_2$  mit den Sequenzlängen  $n$  bzw.  $m$  gegeben. Dann gilt: Ein Edit Transscript für  $S_1$  und  $S_2$  besitzt eine minimale Anzahl von Editieroperationen genau dann, wenn ein Pfad mit insgesamt minimalen Kosten, ausgehend von der Knotenposition  $(0, 0)$  zur Knotenposition  $(n, m)$  im Alignment-Graph  $G_{S_1, S_2}$  korrespondiert.*

Weiter erhält GUSFIELD [99] unmittelbar das

**Corollar 5.5.3** *Die Menge der Pfade mit insgesamt minimalen Kosten von der Knotenposition  $(0, 0)$  zur Knotenposition  $(n, m)$  im Alignment-Graph spezifiziert genau die Menge der optimalen Edit Transscripts von  $S_1$  zu  $S_2$ . Entsprechend beschreibt die Menge der optimalen Edit Transscripts die Menge aller optimalen Alignments von  $S_1$  zu  $S_2$ .*

Zur besseren Veranschaulichung des Alignment-Graphen wird ein Beispiel<sup>18</sup> betrachtet. Dazu sei  $S_1 := v_0 \circ v_1$  und  $S_2 := v_1 \circ v_0 \circ v_1$ .

Der Alignment-Graph  $G_{S_1, S_2}$  ist in Abbildung (5.9) zu sehen. Die fettgedruckten Pfeile geben ein optimales Alignment an. Das Alignment ist dann

<sup>18</sup>Vereinfachend wurden die oberen Indizes der Knoten weggelassen.

$$\begin{array}{ccc} - & v_0 & v_1 \\ v_1 & v_0 & v_1 \end{array}$$

Entsprechend der Kantenmarkierungsfunktion  $f_{E_{S_1, S_2}} : E_{S_1, S_2} \rightarrow \mathbb{R}_+$  wird dann jedem align-ten Paar  $[a, b]$  ein Kostenwert  $d([a, b]) \in \mathbb{R}_+$  zugeordnet. Allgemein betrachtet sind dabei  $a, b$  Zeichen aus den Sequenzen  $S_1$  und  $S_2$  oder das Lückensymbol  $-$ . Definiert man nun die Kosten eines Edit Transscripts der Sequenzen  $S_1$  und  $S_2$  als

- $d(-, S_2[j])$ : Kosten für die Löschung  $S_1[i]$  durch  $S_2[j]$ ,
- $d(S_2[j], -)$ : Kosten für Einfügung von  $S_2[j]$  in  $S_1$ ,
- $d(S_1[i], S_2[j])$ : Kosten für die Ersetzung von  $S_1[i]$  durch  $S_2[j]$ ,

so ist die Editierdistanz der Sequenzen  $S_1$  und  $S_2$  definiert durch

$$d_{fin}(S_1, S_2) := \min_{S_1 \rightarrow S_2} \sum d([a, b]). \quad (5.14)$$

Die Editierdistanz (5.14) drückt damit das Kostenminimum für alle Folgen von Editieroperationen  $[a, b]$  aus, um  $S_1$  in  $S_2$  zu transformieren. Die Gleichung (5.14) ist dabei eine bekannte Wortmetrik, die ursprünglich von LEVENSTEIN [146] entdeckt wurde. In dieser Arbeit wird die LEVENSTEIN-Metrik jedoch in einem neuen Problemkreis angewendet. Sie dient als Hilfsmittel zur Konstruktion optimaler Sequenzalignments, die zur Berechnung der strukturellen Ähnlichkeit graphbasierter Objekte verwendet werden.

Der Algorithmus, der nun das optimale Alignment zwischen  $S_1$  und  $S_2$  mit Hilfe der dynamischen Programmierung berechnet, erzeugt eine Matrix  $(\mathcal{M}(i, j))_{ij}$ ,  $0 \leq i \leq n, 0 \leq j \leq m$ . Es ist dabei  $\mathcal{M}(i, j)$  die Editierdistanz der Sequenzen  $\tilde{S}_1$ ,  $\tilde{S}_2$ , wobei  $\tilde{S}_1$  bzw.  $\tilde{S}_2$  aus den ersten  $i$  Zeichen von  $S_1$  bzw. aus den ersten  $j$  Zeichen von  $S_2$  besteht. Gesucht ist nun ein optimaler und damit minimale Kosten produzierender Pfad von  $\mathcal{M}(0, 0)$  nach  $\mathcal{M}(n, m)$ .  $\mathcal{M}(n, m)$  entspricht gerade der Editierdistanz (5.14). Der eigentliche und bereits bekannte Algorithmus [99, 145] ist rekursiv und wird wie folgt angegeben:

**Definition 5.5.4 (Algorithmus-Editierdistanz)** *Es seien die Sequenzen  $S_1$  bzw.  $S_2$  mit den Wortlängen  $n$  bzw.  $m$  gegeben.*

$$\mathcal{M}(0, 0) := 0, \quad (5.15)$$

$$\mathcal{M}(i, 0) := \mathcal{M}(i-1, 0) + \alpha(S_1[i], -) : 1 \leq i \leq n, \quad (5.16)$$

$$\mathcal{M}(0, j) := \mathcal{M}(0, j-1) + \alpha(-, S_2[j]) : 1 \leq j \leq m, \quad (5.17)$$

$$\mathcal{M}(i, j) := \min \begin{cases} \mathcal{M}(i-1, j) + \alpha(S_1[i], -) \\ \mathcal{M}(i, j-1) + \alpha(-, S_2[j]) \\ \mathcal{M}(i-1, j-1) + \alpha(S_1[i], S_2[j]). \end{cases} : i \in [1, n], j \in [1, m] \quad (5.18)$$

*Dieser Algorithmus berechnet die Editierdistanz  $\mathcal{M}(n, m)$  zwischen  $S_1$  bzw.  $S_2$ . Dabei bezeichnet  $\alpha$  eine Kostenfunktion, die den Editieroperationen die jeweiligen Kosten zuordnet.*

Für die dynamische Programmierung ist typisch, dass sich der Algorithmus rekursiv aus optimalen Teillösungen zusammensetzt. Durch die Bedingung (5.15) wird das Matrixelement  $\mathcal{M}(0, 0)$  initialisiert, die Gleichungen (5.16), (5.17) legen die Lückenstrafe für alle Zeichen der beiden Sequenzen fest. Dazu wird die erste Zeile und die erste Spalte der Matrix erzeugt. Die Bedingung (5.18) sagt aus, dass alle drei möglichen Schritte in die Berechnung eingehen und daraus der Minimalwert gewählt wird. Da nicht nur die Werte  $\mathcal{M}(i, j)$ , sondern auch *Zeiger* zu den entsprechenden Matrixeinträgen gespeichert werden, findet man ein optimales Alignment durch *Traceback*. Das bedeutet, man verfolgt den Weg anhand der Minimumzeiger von  $\mathcal{M}(n, m)$  zu  $\mathcal{M}(0, 0)$  zurück und erhält somit das optimale Alignment. Jedoch muss das optimale Alignment nicht notwendig eindeutig sein.

Im Hinblick auf das web-basierte Graphmatching und deren Anwendungen im Web Structure Mining wird abschließend eine einfache Komplexitätsaussage bezüglich des Algorithmus von Definition (5.5.4) angegeben.

**Proposition 5.5.5** *Es seien die Sequenzen  $S_1$  und  $S_2$  gegeben, denen die Graphen  $\hat{\mathcal{H}}_m^1$  und  $\hat{\mathcal{H}}_m^2$  mit ihren Knotenmengen  $\hat{V}_1$  und  $\hat{V}_2$  zu Grunde liegen. Der Algorithmus (Definition (5.5.4)) zur Bestimmung des optimalen Sequenz-Alignments besitzt eine Komplexität von  $\mathcal{O}(|\hat{V}_1| \cdot |\hat{V}_2|)$ .*

**Beweis:** Nach Definition (5.5.4) und insbesondere mit der Minimumbedingung (5.18) ist klar, dass die Matrix für die dynamische Programmierung mit dem Aufwand  $\mathcal{O}(|\hat{V}_1| \cdot |\hat{V}_2|)$  erzeugt wird. Die Editierdistanz  $\mathcal{M}(n, m)$ <sup>19</sup> besitzt also die Komplexität  $\mathcal{O}(|\hat{V}_1| \cdot |\hat{V}_2|)$ . Mit Corollar (5.5.3) hat man damit auch das optimale Alignment identifiziert.  $\square$

## 5.6 Strukturelle Ähnlichkeit hierarchisierter und gerichteter Graphen

In Kapitel (5.5) wurde auf der Basis der dynamischen Programmierung ein bekannter Algorithmus zur Bestimmung von optimalen Sequenz-Alignments angegeben [145]. Optimale Sequenz-Alignments tragen in dieser Arbeit zur Entwicklung neuartiger und ähnlichkeitsbasierter Analysemethoden im Bereich des

---

<sup>19</sup>Vorausgesetzt  $S_1$  bzw.  $S_2$  besitzen die Wortlängen  $n$  bzw.  $m$ .

Web Structure Mining bei. Da die im Hypertextumfeld bekannten graphentheoretischen Indizes [63, 155] auf Grund ihrer beschränkten Aussagekraft nicht zur ähnlichkeitsbasierten Gruppierung graphbasierter Hypertexte verwendet werden können, besteht ein besonderer Bedarf an Verfahren, welche aussagekräftige und ganzheitliche Graphvergleiche web-basierter Dokumente ermöglichen. Im vorliegenden Kapitel (5.6) wird nun auf der Basis des zentralen Lösungsansatzes aus Kapitel (5.4) die mathematische Konstruktion von Ähnlichkeitsmaßen vorgestellt, die für ganzheitliche Graphvergleiche graphbasierter Hypertexte verwendet werden.

Mit Hilfe von Definition (4.1.10) aus Kapitel (4.1.3) werden zunächst Funktionen eingeführt, die zur Bewertung der Sequenz-Alignments dienen. Prinzipiell können dabei beliebige Funktionen verwendet werden. Aus Gründen der guten Interpretierbarkeit und der Möglichkeit Bewertungsparameter zu verwenden, wird im Folgenden die bekannte GAUSS-Funktion [33] der Form  $e^{-(ax)^2} \in [0, 1]$ ,  $a \in \mathbb{R}$  verwendet, die als *Fensterfunktion* besonders in der Nachrichtentechnik Anwendung findet. Weiterhin tritt die GAUSS-Funktion oft in Untersuchungen der Wahrscheinlichkeitsdichte von Zufallsgrößen auf [33]. Die Basis der GAUSS-Funktion bildet dabei die aus der Analysis bekannte *e*-Funktion, wobei die *konvergente Taylorreihenentwicklung*  $\sum_{i=0}^{\infty} \frac{x^i}{i!}$  von  $e^x$  besteht [33]. Elementare Eigenschaften der GAUSS-Funktion sind z.B.: (i) die *X*-Achse ist *Asymptote* für  $x \rightarrow \infty$  und (ii) die *Y*-Achse bildet die *Symmetrieachse* [33].

Je nachdem ob eine Ähnlichkeits- oder Abstandsfunktion benötigt wird, fällt die finale Bewertungsfunktion auf Basis der GAUSS-Funktion unterschiedlich aus. Wie gezeigt wird, erfüllen die verwendeten Funktionen die Eigenschaften von Definition (4.1.10).

**Lemma 5.6.1** *Es sei  $\omega : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$  definiert durch  $\omega(x, y) := 1 - e^{-\frac{1}{2} \frac{(x-y)^2}{(\sigma)^2}}$ ,  $\sigma \in \mathbb{R}$ . Dann ist  $\omega$  ein Abstandsmaß.*

**Beweis:** Es sind nur die Bedingungen von Definition (4.1.10) nachzuweisen. Dass  $\omega(x, y) \in [0, 1]$ ,  $\forall x, y \in \mathbb{R}$  gilt, folgt unmittelbar aus der Definition von  $\omega$ . Weiter ist  $\omega(x, x) = 1 - 1 = 0$ ,  $\forall x \in \mathbb{R}$ . Da  $(x - y)^2 = (y - x)^2$ ,  $\forall x, y \in \mathbb{R}$ , folgt daraus auch die Symmetriebedingung. Diese Eigenschaften gelten unabhängig vom Parameter  $\sigma \in \mathbb{R}$ .

**Lemma 5.6.2** *Es sei  $\pi : \mathbb{R} \times \mathbb{R} \rightarrow [0, 1]$  definiert durch  $\pi(x, y) := e^{-\frac{1}{2} \frac{(x-y)^2}{(\sigma)^2}}$ ,  $\sigma \in \mathbb{R}$ . Dann ist  $\pi$  ein Ähnlichkeitsmaß.*

**Beweis:** Analog zum Beweis von Lemma (5.6.1).



Um nun konkrete Abstandsmaße als Bewertungsfunktionen einzuführen, um mit deren Hilfe das optimale Alignment zwischen den Sequenzen

$$\begin{aligned} S_1 &:= w_1^{\hat{\mathcal{H}}_m^1} \circ v_{1,1}^{\hat{\mathcal{H}}_m^1} \circ v_{1,2}^{\hat{\mathcal{H}}_m^1} \circ \dots \circ v_{h_1, \sigma_{h_1}}^{\hat{\mathcal{H}}_m^1}, \\ S_2 &:= w_2^{\hat{\mathcal{H}}_m^2} \circ v_{1,1}^{\hat{\mathcal{H}}_m^2} \circ v_{1,2}^{\hat{\mathcal{H}}_m^2} \circ \dots \circ v_{h_2, \sigma_{h_2}}^{\hat{\mathcal{H}}_m^2}, \end{aligned}$$

zu bestimmen, definiere man

$$\alpha^{out} \left( v_{i_1, j_1}^{\hat{\mathcal{H}}_m^1}, v_{i_2, j_2}^{\hat{\mathcal{H}}_m^2} \right) := \begin{cases} \omega^{out} \left( \delta_{out}(v_{i_1, j_1}^{\hat{\mathcal{H}}_m^1}), \delta_{out}(v_{i_2, j_2}^{\hat{\mathcal{H}}_m^2}), \sigma_{out}^1 \right) & : i_1 = i_2 \\ +\infty & : else, \end{cases} \quad (5.19)$$

$0 \leq i_k \leq h_k, 1 \leq j_k \leq \sigma_{i_k}, k \in \{1, 2\}$ . Dabei gilt

$$\omega^{out}(x, y, \sigma_{out}^k) := 1 - e^{-\frac{1}{2} \frac{(x-y)^2}{(\sigma_{out}^k)^2}}, \quad x, y, \sigma_{out}^k \in \mathbb{R},$$

und

$$\alpha^{out} \left( v_{i, j_1}^{\hat{\mathcal{H}}_m^1}, - \right) := \omega^{out} \left( \delta_{out}(v_{i, j_1}^{\hat{\mathcal{H}}_m^1}), \xi, \sigma_{out}^2 \right), \quad (5.20)$$

$$\alpha^{out} \left( -, v_{i, j_2}^{\hat{\mathcal{H}}_m^2} \right) := \omega^{out} \left( \xi, \delta_{out}(v_{i, j_2}^{\hat{\mathcal{H}}_m^2}), \sigma_{out}^2 \right). \quad (5.21)$$

Die modifizierte Definition von  $\omega^{out}$  berührt nicht die in Lemma (5.6.1) gezeigten Eigenschaften, sondern drückt lediglich eine Justierung des Abstandsmaßes auf der Basis von  $\sigma_{out}^k$  aus. Wird jetzt auf der Grundlage von

$$\omega^{in}(x, y, \sigma_{in}^k) := 1 - e^{-\frac{1}{2} \frac{(x-y)^2}{(\sigma_{in}^k)^2}}$$

in analoger Weise die Abstandsfunktion  $\omega^{in}$  definiert als

$$\alpha^{in} \left( v_{i_1, j_1}^{\hat{\mathcal{H}}_m^1}, v_{i_2, j_2}^{\hat{\mathcal{H}}_m^2} \right) := \begin{cases} \omega^{in} \left( \delta_{in}(v_{i_1, j_1}^{\hat{\mathcal{H}}_m^1}), \delta_{in}(v_{i_2, j_2}^{\hat{\mathcal{H}}_m^2}), \sigma_{in}^1 \right) & : i_1 = i_2 \\ +\infty & : else, \end{cases} \quad (5.22)$$

$$\alpha^{in} \left( v_{i, j_1}^{\hat{\mathcal{H}}_m^1}, - \right) := \omega^{in} \left( \delta_{in}(v_{i, j_1}^{\hat{\mathcal{H}}_m^1}), \xi, \sigma_{in}^2 \right), \quad (5.23)$$

$$\alpha^{in} \left( -, v_{i, j_2}^{\hat{\mathcal{H}}_m^2} \right) := \omega^{in} \left( \xi, \delta_{in}(v_{i, j_2}^{\hat{\mathcal{H}}_m^2}), \sigma_{in}^2 \right), \quad (5.24)$$

so sind damit Abstandsmaße zur globalen Bewertung der Alignments<sup>20</sup> bereitgestellt. Somit kann auf der Basis der Kostenfunktion  $\alpha$  das optimale Alignment nach Definition (5.5.4) berechnet werden. Der Parameter  $\xi$  mit  $\xi > 0$  der in den Gleichungen (5.20), (5.21), (5.23), (5.24) auftritt, bewirkt, dass Alignments zwischen zwei Blättern in den jeweiligen Graphen besser bewertet werden als Alignments zwischen einem Blatt und einem Lückensymbol<sup>21</sup>. Weiterhin drücken die obigen Definitionen aus, dass die Alignments nur auf gleichen Ebenen durchgeführt werden. Dies wird mit einer künstlichen Bestrafung  $(+\infty)$  bewirkt, da der Algorithmus, der auf dynamischer Programmierung beruht, niemals diesen kostenintensiven Pfad in der Berechnungsmatrix<sup>22</sup> wählt.

Da das finale Graphähnlichkeitsmaß, wie in Kapitel (5.4) erläutert wurde, ebenenorientiert<sup>23</sup> ist und damit Werte gewünscht sind, die die Ähnlichkeit von Alignments der induzierten Ausgangsgrad- und Eingangsgradsequenzen auf den Ebenen  $i$ ,  $0 \leq i \leq \rho := \max(h_1, h_2)$  detektieren, definiert man in analoger Weise Funktionen zur Bewertung der Ebenen-Alignments. Mit Hilfe einer Abbildung

$$\text{align} \left( v_{i,j_1}^{\mathcal{H}_m^1} \right) := \begin{cases} v_{i,j_2}^{\mathcal{H}_m^2} & : \text{align}^{-1} \left( v_{i,j_2}^{\mathcal{H}_m^2} \right) = v_{i,j_1}^{\mathcal{H}_m^1} \\ - & : \text{else,} \end{cases} \quad (5.25)$$

die einem Knoten im ersten Graph  $v_{i,j_1}^{\mathcal{H}_m^1}$  den beim Traceback ermittelten<sup>24</sup> Knoten  $v_{i,j_2}^{\mathcal{H}_m^2}$  im zweiten Graph zuordnet, kann ein kumulativer, normierter Ähnlichkeitswert für die Ausgangsgrad- und Eingangsgradalignments angegeben werden durch

$$\gamma_{\mathcal{H}_m^k}^{\text{out}}(i) := \frac{\sum_{j=1}^{\sigma_i^k} \hat{\alpha}_{\text{out}} \left( v_{i,j}^{\mathcal{H}_m^k}, \text{align} \left( v_{i,j}^{\mathcal{H}_m^k} \right) \right)}{\sigma_i^k}, \quad (5.26)$$

$$\gamma_{\mathcal{H}_m^k}^{\text{in}}(i) := \frac{\sum_{j=1}^{\sigma_i^k} \hat{\alpha}_{\text{in}} \left( v_{i,j}^{\mathcal{H}_m^k}, \text{align} \left( v_{i,j}^{\mathcal{H}_m^k} \right) \right)}{\sigma_i^k}, \quad (5.27)$$

$k \in \{1, 2\}$ , jeweils aus der Sicht<sup>25</sup> der Sequenz  $S_k$ .  $\hat{\alpha}_{\text{out}}$  bzw.  $\hat{\alpha}_{\text{in}}$  sind in den Gleichungen (5.26), (5.27) völlig analog zu  $\alpha_{\text{out}}$  bzw.  $\alpha_{\text{in}}$  definiert. Die Justierungsparameter werden in gleicher Weise mit Werten  $\hat{\sigma}_{\text{out}}^1, \hat{\sigma}_{\text{out}}^2$  bzw.  $\hat{\sigma}_{\text{in}}^1, \hat{\sigma}_{\text{in}}^2$  belegt.

---

<sup>20</sup>Insgesamt gibt es damit die Alignment-Typen  $[v, u]$ ,  $[v, -]$  bzw.  $[-, v]$ , hier dargestellt in vereinfachter Schreibweise.

<sup>21</sup>Dies wird durchgängig mit '-' gekennzeichnet.

<sup>22</sup>Damit ist die Matrix gemeint, die der DP-Algorithmus erzeugt.

<sup>23</sup>Gemeint ist damit, dass letztlich Ähnlichkeitswerte auf allen Ebenen vorliegen.

<sup>24</sup>Unter minimalen Kosten.

<sup>25</sup>Es besteht ein einfacher Zusammenhang zwischen den Editieroperationen. Wenn eine Sequenz, auf der Basis der bekannten Editieroperationen, in eine andere transformiert wird, so ist eine Einfügung aus der Sicht der einen Sequenz eine Löschung aus der Sicht der anderen.

Damit ist nun die Möglichkeit gegeben, einen kumulierten und normierten Ähnlichkeitswert für die Ausgangsgrad- und Eingangsgradalignments auf der Ebene  $i$  zu bestimmen. Es gilt

$$\begin{aligned} \gamma^{out}(i) := & 1 - \\ & \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{out} \left( v_{i,j}^{\mathcal{H}_m^1}, \text{align} \left( v_{i,j}^{\mathcal{H}_m^1} \right) \right) \right\} + \\ & \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{out} \left( v_{i,j}^{\mathcal{H}_m^2}, \text{align} \left( v_{i,j}^{\mathcal{H}_m^2} \right) \right) \right\}, \end{aligned} \quad (5.28)$$

und entsprechend

$$\begin{aligned} \gamma^{in}(i) := & 1 - \\ & \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{in} \left( v_{i,j}^{\mathcal{H}_m^1}, \text{align} \left( v_{i,j}^{\mathcal{H}_m^1} \right) \right) \right\} + \\ & \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{in} \left( v_{i,j}^{\mathcal{H}_m^2}, \text{align} \left( v_{i,j}^{\mathcal{H}_m^2} \right) \right) \right\}, \end{aligned} \quad (5.29)$$

für  $0 \leq i \leq \rho$ ,  $\rho := \max(h_1, h_2)$ . Im Folgenden wird gezeigt, dass man aus den Gleichungen (5.28), (5.29) Ähnlichkeitsmaße auf hierarchisierten und gerichteten Graphen konstruieren kann, die naheliegende und wohldefinierte Eigenschaften besitzen. Um die so gebildeten Graphähnlichkeitsmaße in eine Klasse von bekannten Ähnlichkeitsmaßen einzuordnen, folgt zunächst eine Definition von BATAGELJ [11], die auf Grund ihrer Bedingungen die Beziehung zwischen Abstand und Ähnlichkeit ausdrückt.

**Definition 5.6.3 (Forward-Backward-Ähnlichkeitsmaß)** *Es sei  $E$  eine Menge von Einheiten, also die strukturelle Beschreibung der Objekte und eine Abbildung  $\phi : E \times E \rightarrow [0, 1]$ . Dann gilt:  $\phi$  heißt Ähnlichkeitsmaß auf der Menge  $E$ , falls die Eigenschaft*

$$\phi(u, v) = \phi(v, u), \forall u, v \in E \quad (\text{Symmetrie}) \quad (5.30)$$

*gilt und entweder die Eigenschaft*

$$\phi(u, u) \leq \phi(u, v), \forall u, v \in E \quad (\text{Forward}) \quad (5.31)$$

*oder*

$$\phi(u, u) \geq \phi(u, v), \forall u, v \in E \quad (\text{Backward}) \quad (5.32)$$

*gilt.*

Das zentrale Ergebnis von Kapitel (5.6) ist, dass die folgenden Ähnlichkeitsmaße bezogen auf die Menge der hierarchisierten und gerichteten Graphen, die Backward-Eigenschaft aus Definition (5.6.3) besitzen.

**Satz 5.6.4** *Es seien die Graphen  $\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2$  gegeben. Weiter sei  $\lambda_i \in \mathbb{R}_+$ ,  $0 \leq i \leq \rho$ ,  $\rho := \max(h_1, h_2)$ . Es gilt:*

$$d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) := \frac{\sum_{i=0}^{\rho} \lambda_i \cdot \gamma^{fin}(i)}{\sum_{i=0}^{\rho} \lambda_i}, \quad (5.33)$$

$$d_2(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) := \frac{\sum_{i=0}^{\rho} \gamma^{fin}(i)}{\rho + 1}, \quad (5.34)$$

$$d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) := \frac{\prod_{i=0}^{\rho} \gamma^{fin}(i)}{d_2(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)}, \quad (5.35)$$

sind Backward-Ähnlichkeitsmaße. Dabei ist  $\gamma^{fin}(i)$  definiert als

$$\gamma^{fin}(i) := \zeta \cdot \gamma^{out}(i) + (1 - \zeta) \cdot \gamma^{in}(i), \quad \zeta \in [0, 1]. \quad (5.36)$$

Um den Satz (5.6.4) zu beweisen, wird ein einfaches Lemma im Voraus formuliert. Es drückt eine bekannte Beziehung zwischen dem arithmetischen und geometrischen Mittel aus.

**Lemma 5.6.5** *Es gilt die Abschätzung*

$$(p_1 \cdot p_2 \cdots p_n)^{\frac{1}{n}} \leq \frac{p_1 + p_2 + \cdots + p_n}{n}, \quad p_i \geq 0, \quad 1 \leq i \leq n. \quad (5.37)$$

**Beweis:** siehe [111].

**Beweis von Satz (5.6.4):** Zunächst werden die behaupteten Eigenschaften für die Maßzahl  $d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  bewiesen, wobei zuerst  $1 \geq d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  zu zeigen ist. Dazu betrachte man die Definition von  $\gamma^{out}(i)$ , also

$$\begin{aligned} \gamma^{out}(i) := & 1 - \\ & \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{out} \left( v_{i,j}^{\hat{\mathcal{H}}_m^1}, \text{align} \left( v_{i,j}^{\hat{\mathcal{H}}_m^1} \right) \right) \right\} + \\ & \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{out} \left( v_{i,j}^{\hat{\mathcal{H}}_m^2}, \text{align} \left( v_{i,j}^{\hat{\mathcal{H}}_m^2} \right) \right) \right\}. \end{aligned}$$

Um den Wertebereich von  $\gamma^{out}(i)$  zu bestimmen, hat man hauptsächlich  $\hat{\alpha}^{out}$  zu betrachten. Nach Definition (analog wie  $\alpha^{out}$ ) treten die Fälle  $\hat{\alpha}^{out}\left(v_{i,j_1}^{\hat{\mathcal{H}}_m^1}, -\right)$ ,  $\hat{\alpha}^{out}\left(-, v_{i,j_2}^{\hat{\mathcal{H}}_m^2}\right)$ ,  $\hat{\alpha}^{out}\left(v_{i,j_1}^{\hat{\mathcal{H}}_m^1}, v_{i,j_2}^{\hat{\mathcal{H}}_m^2}\right)$  auf. Da diese Definitionen auf Funktionen des Typs  $\omega(x, y) := 1 - e^{-\frac{1}{2} \frac{(x-y)^2}{(\sigma)^2}}$  basieren, erhält man zusammen mit Gleichung (5.25)

$$\hat{\alpha}^{out}\left(v_{i,j_1}^{\hat{\mathcal{H}}_m^1}, \text{align}_{out}\left(v_{i,j_1}^{\hat{\mathcal{H}}_m^1}\right)\right) \leq 1 \quad \text{und} \quad \hat{\alpha}^{out}\left(v_{i,j_2}^{\hat{\mathcal{H}}_m^2}, \text{align}_{out}\left(v_{i,j_2}^{\hat{\mathcal{H}}_m^2}\right)\right) \leq 1.$$

Nach Definition von  $\gamma^{out}(i)$  folgt damit die Ungleichung  $\gamma^{out}(i) \leq 1$ . Analog wird auch  $\gamma^{in}(i) \leq 1$  gezeigt.

Da nun  $\gamma^{out}(i) \leq 1$  und  $\gamma^{in}(i) \leq 1$ , bekommt man auch wegen Gleichung (5.36)

$$\gamma^{fin}(i) \leq \zeta + (1 - \zeta) = 1$$

und damit

$$d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \leq \frac{\sum_{i=0}^{\rho} \lambda_i}{\sum_{i=0}^{\rho} \lambda_i} = 1. \quad (5.38)$$

Um die Symmetrieeigenschaft von  $d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  zu zeigen, folgt zunächst die additive Vertauschbarkeit der Glieder von  $\gamma^{out}$  und  $\gamma^{in}$ , also

$$\begin{aligned} \gamma^{out}(i) &:= 1 - \\ &\frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_m^1}, \text{align}\left(v_{i,j}^{\hat{\mathcal{H}}_m^1}\right)\right) \right\} + \\ &\frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_m^2}, \text{align}\left(v_{i,j}^{\hat{\mathcal{H}}_m^2}\right)\right) \right\} \\ &= 1 - \\ &\frac{1}{\sigma_i^2 + \sigma_i^1} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_m^2}, \text{align}\left(v_{i,j}^{\hat{\mathcal{H}}_m^2}\right)\right) \right\} + \\ &\frac{1}{\sigma_i^2 + \sigma_i^1} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{out}\left(v_{i,j}^{\hat{\mathcal{H}}_m^1}, \text{align}\left(v_{i,j}^{\hat{\mathcal{H}}_m^1}\right)\right) \right\} \end{aligned}$$

und analog

$$\begin{aligned}
\gamma^{in}(i) &:= 1 - \\
&\frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{in} \left( v_{i,j}^{\hat{\mathcal{H}}_m^1}, \text{align} \left( v_{i,j}^{\hat{\mathcal{H}}_m^1} \right) \right) \right\} + \\
&\frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{in} \left( v_{i,j}^{\hat{\mathcal{H}}_m^2}, \text{align} \left( v_{i,j}^{\hat{\mathcal{H}}_m^2} \right) \right) \right\} \\
&= 1 - \\
&\frac{1}{\sigma_i^2 + \sigma_i^1} \cdot \left\{ \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^{in} \left( v_{i,j}^{\hat{\mathcal{H}}_m^2}, \text{align} \left( v_{i,j}^{\hat{\mathcal{H}}_m^2} \right) \right) \right\} + \\
&\frac{1}{\sigma_i^2 + \sigma_i^1} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^{in} \left( v_{i,j}^{\hat{\mathcal{H}}_m^1}, \text{align} \left( v_{i,j}^{\hat{\mathcal{H}}_m^1} \right) \right) \right\}.
\end{aligned}$$

Da  $\gamma^{fin}(i)$  nach Gleichung (5.36) additiv definiert ist, folgt jetzt mit der eben gesehenen Vertauschbarkeit von  $\gamma^{out}(i)$  und  $\gamma^{in}(i)$ , dass auch diese Glieder in  $\gamma^{fin}(i)$  vertauschbar sind. Da auch  $d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  nach Definitionsgleichung (5.33) additiv definiert ist, folgt schließlich

$$d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) = d_1(\hat{\mathcal{H}}_m^2, \hat{\mathcal{H}}_m^1).$$

Um den Beweis für das Maß  $d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  abzuschließen, ist noch die Backward-Eigenschaft zu zeigen. Falls nun  $\hat{\mathcal{H}}_m^1 = \hat{\mathcal{H}}_m^2$  gilt, dann ist  $\gamma^{out}(i) = 1$ ,  $\gamma^{in}(i) = 1$  und auch  $\gamma^{fin}(i) = 1$ . Deshalb schließt man aus der Definitionsgleichung (5.33), dass  $d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^1) = 1$  und

$$d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^1) = 1 \geq \frac{\sum_{i=0}^{\rho} \lambda_i \cdot \gamma^{fin}(i)}{\sum_{i=0}^{\rho} \lambda_i} = d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2).$$

Für die Maßzahl  $d_2(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  gibt es nichts zu beweisen, da diese durch die Wahl<sup>26</sup> von  $1 = \lambda_0 = \lambda_1 = \dots = \lambda_{\rho}$  aus  $d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  als Spezialfall hervorgeht.

Um die Aussage des Satzes für die Maßzahl  $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  zu zeigen, kann man Lemma (5.6.5) heranziehen. Da aber  $\gamma^{fin}(i) \leq 1$ , bekommt man insbesondere mit Lemma (5.6.5) die Abschätzung

$$\begin{aligned}
\gamma^{fin}(0) \cdot \gamma^{fin}(1) \cdot \dots \cdot \gamma^{fin}(\rho) &\leq [\gamma^{fin}(0) \cdot \gamma^{fin}(1) \cdot \dots \cdot \gamma^{fin}(\rho)]^{\frac{1}{\rho+1}} \\
&\leq \frac{\gamma^{fin}(0) + \gamma^{fin}(1) + \dots + \gamma^{fin}(\rho)}{\rho + 1}.
\end{aligned}$$

---

<sup>26</sup>In Definitionsgleichung (5.33).

Aus dieser Ungleichungskette folgt aber auch

$$1 \geq \frac{\gamma^{fin}(0) \cdot \gamma^{fin}(1) \cdots \gamma^{fin}(\rho)}{\frac{\gamma^{fin}(0) + \gamma^{fin}(1) + \cdots + \gamma^{fin}(\rho)}{\rho+1}}. \quad (5.39)$$

Die Symmetriebedingung von  $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  ist auf Grund der additiven Vertauschbarkeit von  $\gamma^{fin}(i)$  und der Tatsache, dass der Ausdruck im Nenner gerade  $d_2(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)$  darstellt, erfüllt. Aus Ungleichung (5.39) erhält man nun die Backward-Bedingung

$$1 = d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^1) \geq \frac{\prod_{i=0}^{\rho} \gamma^{fin}(i)}{d_2(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)}.$$

Damit ist der Satz insgesamt bewiesen.  $\square$

Bisher wurden mögliche Knotenmarkierungen der Graphen  $\hat{\mathcal{H}}_m$  bei der Messung der strukturellen Ähnlichkeit nicht berücksichtigt. Damit die strukturelle Ähnlichkeit auch von knotenmarkierten, hierarchisierten und gerichteten Graphen bestimmt werden kann, ist lediglich die Ähnlichkeit zwischen den Knotenmarkierungen einzubeziehen. Dies kann über ein Maß erfolgen, das die Ähnlichkeit der Knotenmarkierungen auf der Grundlage eines Bewertungsschemas misst. Sehr ähnlich zur Definition von  $\gamma^{fin}(i)$ , ausgedrückt durch die Gleichung (5.36), kann nun ein entsprechendes  $\gamma_m^{fin}(i)$  durch

$$\gamma_m^{fin}(i) := (1 - \zeta_m) \cdot \gamma^{fin}(i) + \zeta_m \cdot \gamma_m(i), \quad \zeta_m \in [0, 1], \quad (5.40)$$

definiert werden. Dabei ist

$$\gamma_m(i) := \frac{1}{\sigma_i^1 + \sigma_i^2} \cdot \left\{ \sum_{j=1}^{\sigma_i^1} \hat{\alpha}^m \left( v_{i,j}^{\hat{\mathcal{H}}_1}, \text{align}_m \left( v_{i,j}^{\hat{\mathcal{H}}_1} \right) \right) + \sum_{j=1}^{\sigma_i^2} \hat{\alpha}^m \left( v_{i,j}^{\hat{\mathcal{H}}_2}, \text{align}_m \left( v_{i,j}^{\hat{\mathcal{H}}_2} \right) \right) \right\}.$$

$\hat{\alpha}^m$  misst dabei auf der Basis eines gewählten Bewertungsschemas<sup>27</sup> die Ähnlichkeit zwischen den Knotenmarkierungen. Dabei ist  $\hat{\alpha}^m$ , ähnlich wie die anderen Bewertungsfunktionen, definiert<sup>28</sup>. Konkret wurde dies mit einer Funktion des Typs  $\pi(x, y) := e^{-\frac{1}{2} \frac{(x-y)^2}{(\sigma)^2}}$  realisiert, wobei  $\pi(x, y)$  die Eigenschaften eines Ähnlichkeitsmaßes aus Definition (4.1.10) erfüllt. Auf der Basis von Gleichung (5.40) lässt sich der Satz (5.6.4) in analoger Weise formulieren und beweisen. Abschließend folgt damit das

<sup>27</sup>Dieses wird im konkreten Fall in Form einer positiv reellwertigen Matrix definiert.

<sup>28</sup>Beispielsweise  $\alpha^{out}$  und  $\alpha^{in}$ .

**Corollar 5.6.6** *Es seien die Graphen  $\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2$  und  $\lambda_i \in \mathbb{R}, 0 \leq i \leq \rho, \rho := \max(h_1, h_2)$  gegeben.*

$$d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) := \frac{\sum_{i=0}^{\rho} \lambda_i \cdot \gamma_m^{fin}(i)}{\sum_{i=0}^{\rho} \lambda_i}, \quad (5.41)$$

$$d_2(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) := \frac{\sum_{i=0}^{\rho} \gamma_m^{fin}(i)}{\rho + 1}, \quad (5.42)$$

$$d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) := \frac{\prod_{i=0}^{\rho} \gamma_m^{fin}(i)}{d_2(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2)} \quad (5.43)$$

sind Backward-Ähnlichkeitsmaße.  $\gamma_m^{fin}(i)$  ist auf der Basis der Gleichung (5.40) definiert.

## 5.7 Ergebnisse

In Kapitel (5.6) wurde die Konstruktion des Verfahrens zur Bestimmung der strukturellen Ähnlichkeit von knotenmarkierten, hierarchisierten und gerichteten Graphen vorgestellt. Basierend auf dem Algorithmus zur Berechnung optimaler Sequenz-Alignments gemäß Definition (5.5.4), konnten mit Hilfe von Bewertungsfunktionen die Graphähnlichkeitsmaße  $d_i$  definiert werden. Die so gebildeten Graphähnlichkeitsmaße unterscheiden sich deutlich von Maßen, denen Isomorphiebeziehungen zu Grunde liegen. In den meisten Fällen basieren diese Maße auf dem Prinzip der maximalen Übereinstimmung, das heißt, es werden größte, gemeinsame und isomorphe Untergraphen gesucht. Demgegenüber ist der in dieser Arbeit gewählte Ansatz grundlegend verschieden. Zusammenfassend formuliert, erfolgt zunächst die Transformation der Graphen in formale Knotensequenzen<sup>29</sup>. Darauf basierend wird die Ähnlichkeit der transformierten Strukturen auf Grundlage optimaler Sequenz-Alignments bestimmt.

Folgende elementare Vorteile gegenüber bekannten Ansätzen ergeben sich:

1. Drastische Reduktion der Berechnungskomplexität. Das heißt, dass die Ähnlichkeit der Graphen nun wesentlich effizienter berechnet werden kann, da die Graphen in Form von linearen Sequenzen vorliegen. Damit ist eine wichtige Eigenschaft im Hinblick auf das web-basierte Graphmatching erfüllt.
2. Ein weiterer Vorteil der oben skizzierten Konstruktion ist, dass bei einer noch so komplexen Graphstruktur alle induzierten Ausgangs- und Eingangsgrade in die Berechnung der  $\gamma^{fin}(i)$  einfließen. Dadurch wird die Kantenstruktur während des Graphvergleichs vollständig berücksichtigt.

---

<sup>29</sup>Das sind eindimensionale Strukturen. Siehe Kapitel (5.4).



3. Aus der Parametrisierungsmöglichkeit<sup>30</sup> folgt, dass eine hohe Flexibilität hinsichtlich zu messender Strukturaspekte besteht. Die Parametrisierung wird dabei in zwei Klassen unterteilt:

- (a)  $\sigma_{out}^1, \sigma_{out}^2, \sigma_{in}^1, \sigma_{in}^2 \in \mathbb{R}$  sind die Parameter (global) der Bewertungsfunktionen zur Bestimmung des optimalen Alignments der Grundsequenzen. Dagegen steuern die Parameter (lokal)  $\hat{\sigma}_{out}^1, \hat{\sigma}_{out}^2, \hat{\sigma}_{in}^1, \hat{\sigma}_{in}^2 \in \mathbb{R}$  die Alignment-Bewertung bezüglich der Ebenen.
- (b) Der Parameter  $\zeta \in [0, 1]$  gewichtet Ausgangsgrad- und Eingangsgradalignments, ausgedrückt durch die Gleichung (5.36).

Mit einer speziellen Wahl der Parameter aus Punkt (3a) der Aufzählung, kann das Maß  $d_i$  „härter“ oder „weicher“ eingestellt werden. Ist beispielsweise ein Testkorpus  $T$  gegeben, das Graphen mit stark variierenden Ordnungen enthält, so ist es erforderlich, die eventuell starken „Höhenunterschiede“<sup>31</sup> durch geeignete Wahl dieser Parameter zu bewerten. Bezogen auf Punkt (3b), kann durch die konkrete Wahl von  $\zeta \in [0, 1]$  die Bewertung struktureller Teilaspekte erfolgen, z.B. die ausschließliche Betrachtung der Wurzelbaumstruktur. Daher wirken sich diese Parametrisierungsoptionen in der Anwendung auf neue Problemstellungen positiv aus, da sich das Verhalten des gewünschten Maßes  $d$  für jedes spezielle Graphähnlichkeitsproblem unterscheiden soll.

4. Die in Kapitel (5.6) beschriebene Konstruktion beruht nicht auf Isomorphiebeziehungen der betrachteten Graphen. Bei der Anwendung von Methoden des exakten Graphmatchings [129, 213, 214, 254] besteht oft das Problem, dass nicht isomorphe Graphen, die trotzdem intuitiv ähnlich erscheinen, als weniger ähnlich bewertet werden.

Die in Kapitel (5.6) beschriebene Konstruktion führt jedoch zu einem gewissen Strukturverlust, da auf der Basis der gewählten Alignment-Bewertung die Berücksichtigung der Kantentypen nicht explizit erfolgt. Das bedeutet genauer: Ausgehend von einem existierenden Ausgangsgrad- und Eingangsgradalignment auf der Ebene  $i$  wird nicht berücksichtigt, welcher Kantentyp, z.B. eine Kernel-Kante, eine Across-Kante oder eine Up-Kante, einen gewissen Ausgangs- bzw. Eingangsgrad induziert hat. Da wie beschrieben die Kantentypen nicht explizit berücksichtigt werden, erfolgt als Gegenmaßnahme die jeweilige Berechnung der prozentualen Verteilung der Kantentypen. Damit ist die Möglichkeit gegeben, eine Aussage über die Vorkommenshäufigkeit der Kantentypen zu treffen und damit die Parametrisierung hinsichtlich zu betonender Strukturaspekte passender

---

<sup>30</sup>Parameter der Bewertungsfunktionen.

<sup>31</sup>Bezogen auf zwei Graphenhöhen  $h_i$  und  $h_j$ .

zu wählen. Die experimentellen Ergebnisse aus Kapitel (5.8) zeigen jedoch, dass die Vernachlässigung der Kantentypen keine negativen Auswirkungen auf das Strukturerkennungsverhalten des letztlich verwendeten Graphähnlichkeitsmaßes hat. Insbesondere zeigt Kapitel (5.8.2), dass die auf der Basis von  $d_3$  erzeugte Ähnlichkeitsmatrix zur strukturorientierten Filterung in Form eines Clustering-Experiments geeignet ist. Dies drückt sich in dieser Arbeit durch eine positive Evaluierung der Clustergüte aus. In Kapitel (6) stellt sich weiter heraus, dass das Maß  $d_3$  zur Strukturerkennung komplexer Graphmengen auf großen Datenbeständen einsetzbar ist.

In dieser Arbeit wird die Graphstruktur web-basierter Dokumente als knotenmarkierter, hierarchisierter und gerichteter Graph aufgefasst, wobei in Kapitel (3.2) bereits die Berechnungsvorschrift der Kantenstruktur erklärt wurde. Auf der Basis einer einfachen Heuristik [94, 95] wird ausgehend von einem fest gewählten Startknoten<sup>32</sup> mit Hilfe einer Breitensuche die Kernel-Hierarchie<sup>33</sup> bestimmt. Dabei spiegelt die Kernel-Hierarchie die vom Hypertextautor beabsichtigte Navigationsstruktur der Website wider. Zum einen ist die Berechnung der Kernel-Hierarchie entscheidend für die Konstruktion des Graphähnlichkeitsmodells aus Kapitel (5), da sich aus der Kenntnis der Kernel-Kanten die Ebenendarstellung der Graphstruktur ergibt. Zum anderen besitzt die Berechnungsvorschrift auf Grundlage der Breitensuche einen Schwachpunkt [94]: Down-Kanten werden nicht als solche erkannt, sondern als Kernel-Kanten interpretiert. Zusammen mit den experimentellen Ergebnissen aus Kapitel (5.8) kann jedoch die gewählte Heuristik als gute Approximation an die tatsächlich hierarchische Website-Struktur betrachtet werden. Abschließend für das Kapitel (5.7) sei noch ein Vorteil der hierarchischen Graphdarstellung erwähnt. Die Berechnung der Kernel-Hierarchie und die anschließende Bestimmung der übrigen Kantentypen<sup>34</sup> lässt eine semantische Bewertung der Kantentypen zu. Zum Beispiel können Across-Kanten oft als *Themenwechsel* interpretiert werden, wobei Up-Kanten meistens zur einfacheren Navigation bezüglich Webseiten höhergelegener Ebenen dienen. Zusammen mit der Verteilung der Kantentypen, gibt die hierarchische Graphstruktur einen guten Überblick über potenzielle Navigationsmöglichkeiten.

## 5.8 Experimentelle Ergebnisse

Im folgenden Kapitel (5.8.1) werden die Graphähnlichkeitsmaße aus Kapitel (5.6) auf der Basis des Testkorpus  $T_C$ <sup>35</sup> evaluiert, wobei dessen graphentheoretische

---

<sup>32</sup>Dieser ist immer die Start-Webseite der gesamten Website-Struktur.

<sup>33</sup>Siehe Kapitel (5.3).

<sup>34</sup>Siehe Definition (5.3.1) aus Kapitel (5.3).

<sup>35</sup>Siehe Kapitel (3.2).

Kerndaten	Wert
$\min( \hat{V} )$	5
$\max( \hat{V} )$	97
$\min(\text{diam}(\hat{\mathcal{H}}))$	1
$\max(\text{diam}(\hat{\mathcal{H}}))$	27
$\text{avg}( \hat{V} )$	23
$\text{avg}(\text{diam}(\hat{\mathcal{H}}))$	3

**Abbildung 5.10:** Kerndaten des Graphkorpus  $T_C$ .

Kerndaten in Abbildung (5.10) zu finden sind. Da die Ähnlichkeitsmaße zur strukturellen Untersuchung web-basierter Dokumente eingesetzt werden, liegt der Anwendungsschwerpunkt besonders im Web Structure Mining, da dieses die Erforschung struktureller Eigenschaften hypertextueller Dokumente zum Ziel hat.

Kapitel (5.8.1) untersucht zunächst die Wertebereichsausschöpfung der Graphähnlichkeitsmaße  $d_i$ . Damit kann im Vorfeld einer konkreten Evaluierung ermittelt werden, ob ein bestimmtes Maß  $d_i$  für die fokussierte Anwendung besonders geeignet bzw. ungeeignet ist. Dagegen wird in Kapitel (5.8.2) die Evaluierung web-basierter Dokumente diskutiert, die durch DOM-Strukturen [45] repräsentiert werden. Die Kapitel (5.8.1), (5.8.2) folgen dabei einer gemeinsamen Reihenfolge von Untersuchungsschritten:

- Die Bestimmung einer Ähnlichkeitsmatrix  $(s_{ij})_{ij}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq n$ ,  $s_{ij} \in [0, 1]$ , wobei diese als Grundlage verschiedener Anwendungen zu sehen ist, z.B. zur Berechnung von Verteilungen oder als Basis für den Einsatz von Data Mining-Verfahren.
- Die Anwendung von multivariaten Analyseverfahren, z.B. die Clusteringverfahren.

Die Interpretation der Ergebnisse, zum einen bezogen auf Website-Strukturen und zum anderen auf DOM-Trees, ist jedoch unterschiedlich, da für beide Dokumentengruppen verschiedene Problemstellungen behandelt werden.

### 5.8.1 Experimente mit Website-Strukturen

Um nun die Graphähnlichkeitsmaße aus Satz (5.6.4) auf der Basis von  $T_C$  zu evaluieren, ist der erste Schritt die Untersuchung der Wertebereichsausschöpfung. Genauer ist dabei zu bestimmen, wie gut die Maße

$$\begin{aligned}
d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) &:= \frac{\sum_{i=0}^{\rho} \lambda_i \cdot \gamma^{fin}(i)}{\sum_{i=0}^{\rho} \lambda_i}, \\
d_2(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) &:= \frac{\sum_{i=0}^{\rho} \gamma^{fin}(i)}{\rho + 1}, \\
d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) &:= \frac{\prod_{i=0}^{\rho} \gamma^{fin}(i)}{\frac{\sum_{i=0}^{\rho} \gamma^{fin}(i)}{\rho+1}},
\end{aligned}$$

ihren Wertebereich ausschöpfen. Dabei gilt nach Konstruktion  $d_i(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \in [0, 1]$ ,  $i = 1, 2, 3$ . Für diese Untersuchung werden einige Fälle unterschieden, wobei sich darin unterschiedliche Parameterbelegungen und die Betonung verschiedenartiger struktureller Aspekte widerspiegeln. Dazu wurden die folgenden Datenklassen definiert:

**Definition 5.8.1** *Die Datenklassen  $D_1$ - $D_5$ , die durch unterschiedliche Parameterbelegungen definiert werden, beziehen sich auf das Testkorpus  $T_C$ :*

1.  $D_1$ :  $\zeta = 1.0$  (Ausschließlich Alignments von Kernel-Kanten); Falls  $\gamma^{fin}(i) < 0.5$ , setze  $\lambda_i = 100$ , andernfalls  $\lambda_i = 1$ ; Parameterbelegung:

$$\begin{aligned}
\sigma_{out}^1 &= 1.0, \sigma_{out}^2 = 2.0, \sigma_{in}^1 = 1.0, \sigma_{in}^2 = 2.0, \hat{\sigma}_{out}^1 = 3.0, \hat{\sigma}_{out}^2 = 5.0, \\
\hat{\sigma}_{in}^1 &= 3.0, \hat{\sigma}_{in}^2 = 5.0.
\end{aligned}$$

2.  $D_2$ :  $\zeta = 0.3$ ; Falls  $\gamma^{fin}(i) < 0.5$ , setze  $\lambda_i = 100$ , andernfalls  $\lambda_i = 1$ ; Parameterbelegung:

$$\begin{aligned}
\sigma_{out}^1 &= 1.0, \sigma_{out}^2 = 1.0, \sigma_{in}^1 = 1.0, \sigma_{in}^2 = 1.0, \hat{\sigma}_{out}^1 = 1.0, \hat{\sigma}_{out}^2 = 5.0, \\
\hat{\sigma}_{in}^1 &= 1.0, \hat{\sigma}_{in}^2 = 5.0.
\end{aligned}$$

3.  $D_3$ :  $\zeta = 0.5$ ; Falls  $\gamma^{fin}(i) < 0.5$ , setze  $\lambda_i = 100$ , andernfalls  $\lambda_i = 1$ ; Parameterbelegung:

$$\begin{aligned}
\sigma_{out}^1 &= 1.0, \sigma_{out}^2 = 1.0, \sigma_{in}^1 = 1.0, \sigma_{in}^2 = 1.0, \hat{\sigma}_{out}^1 = 1.0, \hat{\sigma}_{out}^2 = 5.0, \\
\hat{\sigma}_{in}^1 &= 1.0, \hat{\sigma}_{in}^2 = 5.0.
\end{aligned}$$

4.  $D_4$ :  $\zeta = 0.5$ ; Falls  $\gamma^{fin}(i) < 0.5$ , setze  $\lambda_i = 64$ , andernfalls  $\lambda_i = 16$ ; Parameterbelegung:

$$\begin{aligned}
\sigma_{out}^1 &= 1.0, \sigma_{out}^2 = 2.0, \sigma_{in}^1 = 1.0, \sigma_{in}^2 = 2.0, \hat{\sigma}_{out}^1 = 3.0, \hat{\sigma}_{out}^2 = 5.0, \\
\hat{\sigma}_{in}^1 &= 3.0, \hat{\sigma}_{in}^2 = 5.0.
\end{aligned}$$

5.  $D_5$ :  $\zeta = 0.5$ ; Falls  $\gamma^{fin}(i) < 0.5$ , setze  $\lambda_i = 100$ , andernfalls  $\lambda_i = 1$ ;  
Parameterbelegung:

$$\sigma_{out}^1 = 1.0, \sigma_{out}^2 = 1.0, \sigma_{in}^1 = 1.0, \sigma_{in}^2 = 1.0, \hat{\sigma}_{out}^1 = 3.0, \hat{\sigma}_{out}^2 = 5.0, \\ \hat{\sigma}_{in}^1 = 3.0, \hat{\sigma}_{in}^2 = 5.0.$$

Zunächst erfolgt für alle Datenklassen der Definition (5.8.1) die Interpretation der Verteilungen aus Abbildung (5.11). Dabei bezieht sich die Anordnung<sup>36</sup> der Verteilungen auf die Datenklassen  $D_1$ - $D_5$ , wobei die jeweilige Parameterbelegung in Definition (5.8.1) angegeben ist. Da nach Satz (5.6.4) die Eigenschaften

$$d_i(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) = d_i(\hat{\mathcal{H}}_m^2, \hat{\mathcal{H}}_m^1) \quad \text{und} \quad d_i(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^1) = 1, \quad \hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2 \in T_C$$

gelten, entsteht die zu Grunde liegende Ähnlichkeitsmatrix durch  $\frac{|T_C|(|T_C|-1)}{2}$ -male Ähnlichkeitsberechnung der Graphpaare, für jedes Maß  $d_i$ . Die Abbildung (5.11) zeigt die gerankten Ähnlichkeitswerte der Maße  $d_1$ ,  $d_2$  und  $d_3$ . Das bedeutet, dass die sortierten Ähnlichkeitspaarungen gegen eine Platznummer aufgetragen wurden.

Aus Abbildung (5.11) ersieht man für alle Datenklassen, dass die Maße

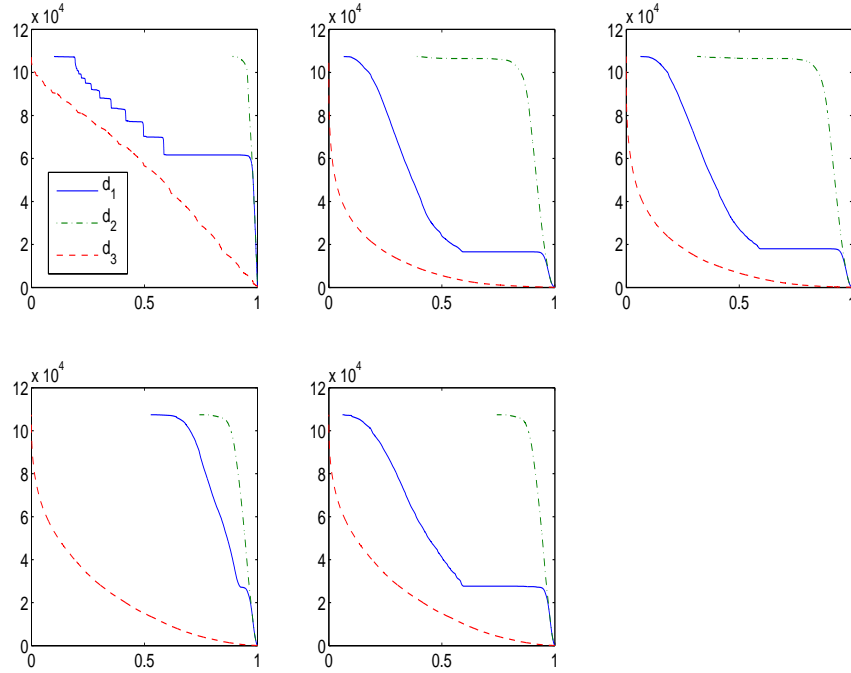
$$d_1(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) := \frac{\sum_{i=0}^{\rho} \lambda_i \cdot \gamma^{fin}(i)}{\sum_{i=0}^{\rho} \lambda_i} \quad \text{und} \quad d_2(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) := \frac{\sum_{i=0}^{\rho} \gamma^{fin}(i)}{\rho + 1},$$

den Wertebereich  $[0, 1]$  nicht vollständig ausschöpfen und die graphbasierten Hypertexte „zu ähnlich“ bewerten. Am deutlichsten erkennt man dies am Verhalten von  $d_2$ . Das liegt zum einen an der Definition von  $d_1$  und  $d_2$ , zum anderen an der Parametrisierung, bezogen auf  $\lambda_i \in \mathbb{R}$ . Da  $d_2$  lediglich das arithmetische Mittel der  $\gamma^{fin}(i)$  ausdrückt, ist die mangelnde Ausschöpfung des Wertebereichs  $[0, 1]$  deutlich ausgeprägter als bei  $d_1$ . Weiterhin reagiert  $d_1$  wesentlich empfindlicher auf Strukturunterschiede als  $d_2$ , wobei sich diese Eigenschaft in den sprunghaften Schaubildern von  $d_1$  widerspiegelt.

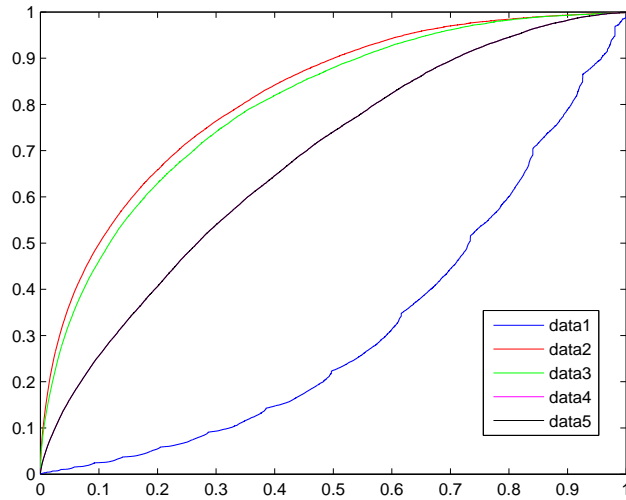
Dagegen zeigt das Maß  $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) := \frac{\sum_{i=0}^{\rho} \gamma^{fin}(i)}{\frac{\sum_{i=0}^{\rho} \gamma^{fin}(i)}{\rho+1}}$  für alle Klassen der Definition (5.8.1) eine gute und kontinuierliche Wertebereichsausschöpfung. Der Kurvenverlauf von  $d_3$  bezogen auf  $D_1$ , unterscheidet sich deutlich von den  $d_3$ -Verläufen der übrigen Klassen. Dieser Unterschied ist dadurch erklärbar, dass in  $D_1$  ausschließlich Alignments von Kernel-Kanten betrachtet wurden. Das bedeutet: Die hypertextuellen Dokumentstrukturen repräsentieren gerichtete Wurzelbäume. Damit wurden die Alignments von Across-Kanten, Up-Kanten und Down-Kanten

---

<sup>36</sup>Das erste Bild in der oberen Reihe aus Abbildung (5.11) bezieht sich auf Datenklasse  $D_1$ , das zweite Bild in der oberen Reihe bezieht sich auf Datenklasse  $D_2$  etc.



**Abbildung 5.11:** Verteilungen der gerankten Ähnlichkeitswerte für  $d_1, d_2$  und  $d_3$ , bezogen auf alle Datenklassen der Definition (5.8.1). Die X-Achse bezeichnet den Ähnlichkeitswert  $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \in [0, 1]$ . Auf der Y-Achse ist die Anzahl der Graphpaare aufgetragen.



**Abbildung 5.12:** Kummulative Ähnlichkeitswertverteilungen für die Datenklassen der Definition (5.8.1). X-Achse: Ähnlichkeitswert  $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \in [0, 1]$ . Y-Achse: Prozentsatz der Graphen, die einen Ähnlichkeitswert  $d_3 \leq X$ -Wert besitzen.

bewusst vernachlässigt. Bezüglich  $D_1$ - $D_4$  wurde der Parameter  $\zeta \in [0, 1]$ , der die Gewichtung der Werte  $\gamma^{out}(i)$  und  $\gamma^{in}(i)$  steuert, variiert.

Abschließend ist das durch die Abbildung (5.11) gezeigte Verhalten von  $d_3$  als positiv zu bewerten. Damit ist eine wesentliche Voraussetzung erfüllt, die die Grundlage für einen sinnvollen Einsatz im Web Structure Mining bildet. Um nun etwas über die Verteilung der Ähnlichkeitswerte innerhalb der Datenklassen zu erfahren, betrachte man die kummulative Ähnlichkeitswertverteilung in Abbildung (5.12). Als Graphähnlichkeitsmaß wurde dabei ausschließlich  $d_3$  verwendet. Auffällig ist, dass sich das Schaubild der Klasse  $D_1$  von den Kurvenverläufen der übrigen Datenklassen prinzipiell unterscheidet. Daraus erkennt man, dass z.B. ca. 20% der Graphen bereits einen Ähnlichkeitswert  $d_3 \leq 0.5$  haben. Im Gegensatz dazu besitzen ca. 90% der Graphen aus  $D_2$  einen Ähnlichkeitswert  $d_3 \leq 0.5$ . Insgesamt ersieht man aus Abbildung (5.12), dass die Graphen in  $D_1$  signifikant ähnlicher<sup>37</sup> bewertet werden als die Graphen der übrigen Datenklassen. Dieses Verhalten ist deshalb plausibel, da die Graphen in  $D_1$  ausschließlich als Wurzelbäume behandelt werden und somit die für Hypertexte typischen Across-Kanten, Up-Kanten und Down-Kanten fehlen. Daraus folgt, dass sich der Großteil dieser Graphen deutlich weniger stark strukturell unterscheidet als die Graphen in den restlichen Datenklassen. Umgekehrt ist die Lage bei  $D_2$ - $D_5$ : Durch Einbeziehung aller Kantentypen gilt der Hauptanteil der Graphen untereinander als strukturell unähnlich. Die Schaubilder für  $D_4$  und  $D_5$  sind identisch.

Um nun zu ermitteln, wie gut das Maß  $d_3$  die strukturelle Ähnlichkeit von web-basierten Hypertexten wiedergibt, wird im Folgenden die Anwendung von Clusteringverfahren fokussiert. Jedoch lässt sich das Analyseergebnis nur für eine wesentlich kleinere Teilmenge  $T_{\text{small}} \subseteq T_C$  anschaulich<sup>38</sup> darstellen. Für dieses Experiment wurde ein agglomeratives Clusteringverfahren gewählt, dessen Funktionsweise Kapitel (2.4.2) erläutert. Ausgehend von  $T_C$  wurde  $T_{\text{small}}$  so erzeugt, dass die Ähnlichkeitswerte der Graphpaarungen mit annähernd gleicher Anzahl vorkommen und den Wertebereich  $[0, 1]$  nahezu vollständig ausschöpfen. Wählt man nun für die Teilmenge  $T_{\text{small}} \subseteq T_C$ ,  $|T_{\text{small}}| = 22$ , zur Berechnung der Fusionschritte den Clusterabstand<sup>39</sup>

$$\alpha_{AL}(C_i, C_j) := \frac{1}{|C_i||C_j|} \sum_{\tilde{\mathcal{H}} \in C_i} \sum_{\mathcal{H} \in C_j} d_3(\mathcal{H}, \tilde{\mathcal{H}}),$$

so erhält man als Ergebnis der Clusterung Abbildung (5.13).

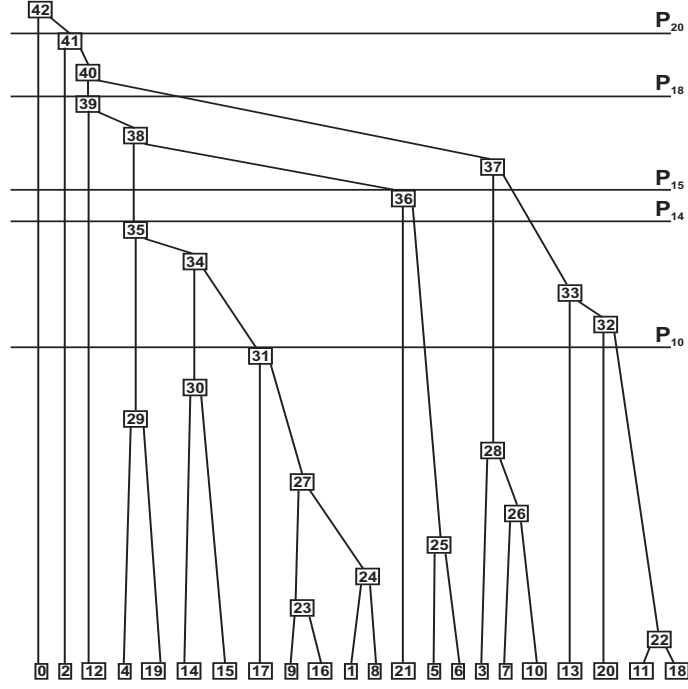
Da nun eine mögliche Clusterlösung interpretiert werden muss, liegt die Fragestellung nahe, ob sich Partitionen angeben lassen, die als mögliche Abbruchstufen gel-

---

<sup>37</sup>Auf der Basis von  $d_3$ .

<sup>38</sup>In Form eines Dendogramms. Siehe Kapitel (2.4.2).

<sup>39</sup>Wegen seiner guten Interpretierbarkeit wurde Average Linkage gewählt. Siehe Kapitel (2.4.2).



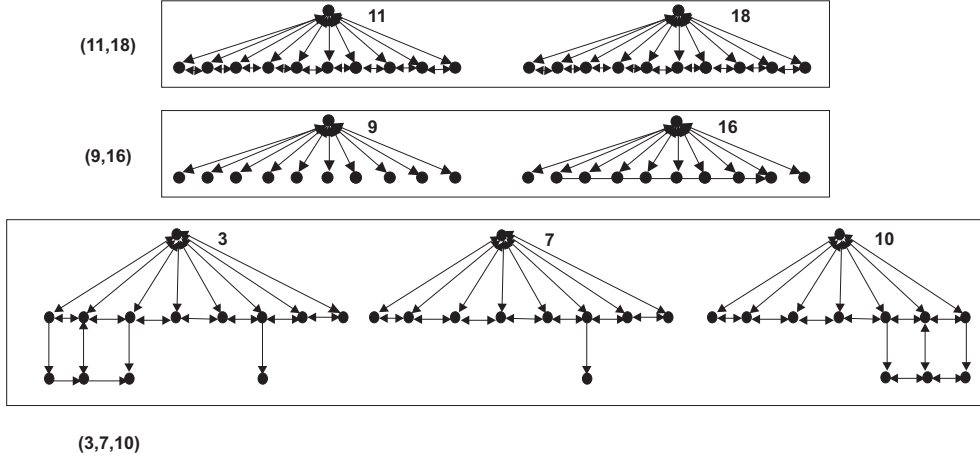
**Abbildung 5.13:** Dendrogramm als Ergebnis der Clusterung von  $T_{\text{small}}$ . Die zweiundzwanzig Graphen sind jeweils mit Objektnummern bezeichnet.

ten können. Ein solches Abbruchkriterium für agglomerative Clusteringverfahren wurde von RIEGER [180] entwickelt. Im Folgenden wird der Clusterabstand auf der  $j$ -ten Fusionsstufe vereinfachend als  $\alpha_{AL}^j$  bezeichnet. RIEGER bildet zunächst die Betragsdifferenzen  $\eta_i = |\alpha_{AL}^j - \alpha_{AL}^{j+1}|$ ,  $i = 1, 2, \dots, m - 2$ , wobei für jeden Fusionschritt  $j$  eine Knotennummer  $m = |T_{\text{small}}| + j$  gebildet wird. Nun kann für jeden weiteren Fusionschritt  $j + 1$  die jeweilige Betragsdifferenz der Clusterabstände berechnet werden, wobei  $j = 2, 3, \dots, m - 1$  gilt. Auf der Basis von  $\bar{\eta} = \frac{1}{m-2} \sum_{i=1}^{m-2} \eta_i$  und der Bildung der Standardabweichung

$$\sigma = \sqrt{\sum_{i=1}^{m-2} (\eta_i - \bar{\eta})^2},$$

definiert RIEGER die untere Schranke  $\theta = \bar{\eta} + \frac{\sigma}{2}$ . Gilt nun  $\eta_i \geq \theta$ , so drückt diese Ungleichung eine gute Trennbarkeit der Cluster aus. Damit gilt jede gebildete Betragsdifferenz, die die Ungleichung  $\eta_i \geq \theta$  erfüllt, als denkbare Abbruchstufe. Das Ergebnis der Berechnung aller möglichen Abbruchstufen ist in Abbildung (5.13) durch horizontale Linien angedeutet. Neben einer guten Trenneigenschaft werden aber auch diejenigen Cluster einer Partition gesucht, die möglichst homogen sind, das heißt, deren Elemente sich auf der Basis des Graphabstandsmaßes  $d_3$  sehr ähnlich sind. Um die Homogenität der Cluster zu beschreiben wurde das





**Abbildung 5.14:** Web-basierte Hypertextgraphen der Cluster (11,18), (9,16) und (3,7,10). Die Clusterschreibweise in Tupelform verwendet dabei die Objektnummern. Es gilt  $\zeta = 0.5$ ; Parameterbelegung siehe Punkt (5) der Definition (5.8.1).

Maß

$$h(C_i) := \frac{1}{|C_i| \cdot (|C_i| - 1)} \sum_{\mu \in I_{C_i}} \sum_{\nu \in I_{C_i}} s_{\mu\nu}$$

angewendet, welches bereits zur Cluster-Interpretation in Kapitel (2.4.1) vorgestellt wurde. Dabei wurde  $s = d_3$  gesetzt;  $C_i$  bezeichnet ein Cluster auf einer gewissen Partition und  $I_C$  die entsprechende Indexmenge. Auf Grund der in Kapitel (2.4.2) erklärten Konstruktion eines agglomerativen Clusteringverfahrens nimmt die Homogenität im Dendrogramm von den Blättern bis zur Wurzel hin immer weiter ab. Bezeichnet nun  $P = (C_1, C_2, \dots, C_k)$  die Clustermenge einer Partition  $P$ , so kann die Einbeziehung der Homogenitätssumme

$$\sum_{i=1}^k h(C_i),$$

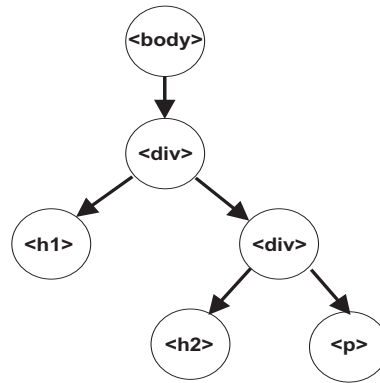
zusammen mit den berechneten Abbruchstufen als mögliches Kriterium für eine Clusterlösung aufgefasst werden. In Abbildung (5.13) wurde die Partition  $P_{10}$  gewählt, da einerseits das Abbruchkriterium erfüllt ist und andererseits die Partition die höchste verbleibende Homogenitätssumme aufweist.

Um einen Eindruck zu bekommen, wie gut  $d_3$  die Ähnlichkeit der Graphen in den Clustern wiedergibt, sei dazu Abbildung (5.14) betrachtet. Aus der Partition  $P_{10}$  wurden beispielhaft die Cluster mit den Graphen (11,18), (9,16) und (3,7,10) ausgewählt. Gemäß der Haupteigenschaft des vorliegenden Clusteringverfahrens enthält das Cluster auf der ersten Fusionsstufe auf der Basis von  $d_3$  die ähnlichsten Graphen: In diesem Fall besitzen die web-basierten Dokumente sogar identische Graphstrukturen. Die Graphen, die das Cluster (9,16) bilden,

```

<body>
  <div style="...">
    <h1>Ueberschrift</h1>
    <div>
      <h2>Kapitel</h2>
      <p>
        Text
      </p>
    </div>
  </div>
</body>

```



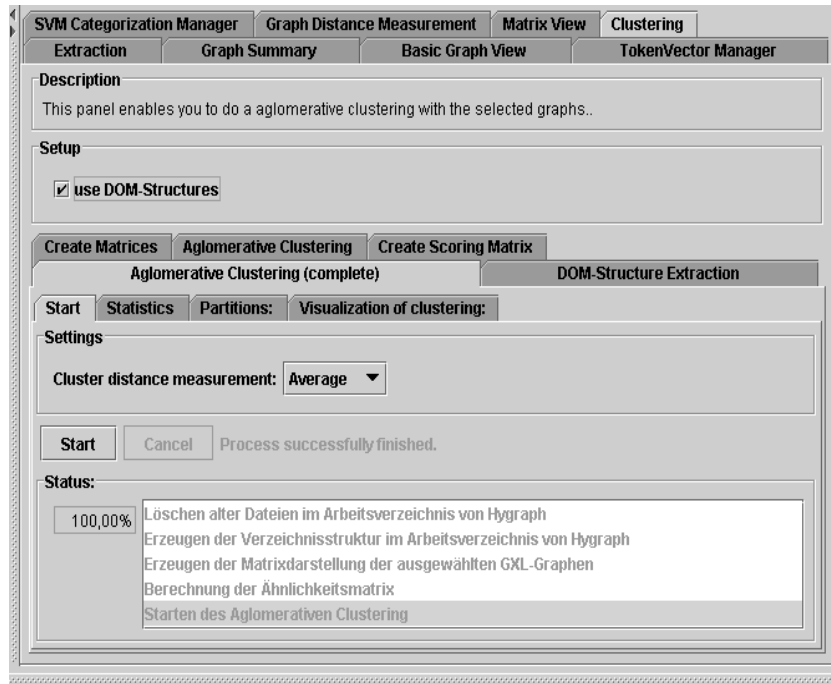
**Abbildung 5.15:** (i): Das linke Bild zeigt ein einfaches HTML-Codefragment. (ii): Das rechte Bild bildet den entsprechenden DOM-Tree ab.

wurden im zweiten Fusionsschritt zusammengeschlossen. Sie unterscheiden sich lediglich durch eine Across-Kante auf der ersten Ebene. Ansonsten stimmen sie in ihren Ordnungen und Kantenmengen vollständig überein. Das Cluster mit den Graphen (3,7,10) wurde in einer fortgeschrittenen Fusionsstufe erzeugt, wobei das agglomerative Verfahren zuerst das Cluster (7,10) bildete. In einem weiteren Fusionsschritt wurde dann der Graph mit der Objektzahl 3 dazugefügt. Die Graphen 3, 7 besitzen bis zur ersten Ebene eine identische Struktur. Strukturelle Unterschiede zeigen sich auf der Ebene 2. Verglichen mit Graph 3 und Graph 7 besitzt Graph 10 bis zur ersten Ebene dieselbe Kantenstruktur, jedoch um einen Knoten reduziert. Für die weitere Interpretation der Cluster auf höher liegenden Partitionen gilt, dass nach Konzeption des agglomerativen Verfahrens die Graphen innerhalb der Cluster immer unähnlicher<sup>40</sup> werden. Im Cluster, welches die Wurzel des Dendogramms repräsentiert, sind schließlich alle Graphen verschmolzen. Es bleibt nun die Aufgabe, die Güte der Strukturerkennung von  $d_3$  auf einen großen Datenbestand zu untersuchen. Dazu wird in Kapitel (6) gezeigt, dass  $d_3$  in der Lage ist, Graphmengen web-basierter Dokumente strukturell zu trennen.

## 5.8.2 Experimente mit web-basierten Dokumenten

In Kapitel (5.8.1) wurde als erster Analyseschritt die Wertebereichsausschöpfung der Graphmaße aus Satz (5.6.4) untersucht. Bezüglich der Wertebereichsausschöpfung und der Cluster-Interpretation von Website-Strukturen erwies sich das Maß  $d_3$  in Kapitel (5.8.1) als sehr geeignet. Auf Grund der Konzeption der Graphmaße können während der Ähnlichkeitsmessung durch die Parametrisierung unterschiedliche Strukturaspekte berücksichtigt werden. Diese Flexibilität wirkt sich

<sup>40</sup>Auf der Basis von  $d_3$ .



**Abbildung 5.16:** Ausgehend von einer Menge von GXL-Repräsentationen erfolgt die web-basierte Extraktion der DOM-Trees mit Hilfe von HTMLParser [257]. Daran anschließend erzeugt HyGraph die jeweilige Ähnlichkeitsmatrix, die als Eingabe des Clusteringverfahrens dient. Die Registerkarten *Statistics* und *Partitions* dienen zur Berechnung der Clusterhomogenität und des Abbruchkriteriums aus Kapitel (5.8.1). Dagegen wird mit Hilfe der Registerkarte *Visualization of Clustering* das entsprechende Dendrogramm erzeugt.

unmittelbar positiv auf die Messung der strukturellen Ähnlichkeit web-basierter Dokumente in Form von DOM-Strukturen [45] aus, da lediglich die Belegung des Parameters  $\zeta_m$  in Gleichung (5.40) zu ändern ist. Bezogen auf die Ähnlichkeitsmessung stellen DOM-Strukturen gerichtete Wurzelbäume mit Knotenmarkierungen dar, wobei die Knotenmarkierungen hier die Bedeutungen der HTML-Tags wiedergeben.

Beispielhaft zeigt die Abbildung (5.15) ein HTML-Codefragment zusammen mit dem zugehörigen DOM-Tree. Um nun die strukturelle Ähnlichkeit dieser Strukturen zu bestimmen, müssen außer einer geeigneten Parametrisierung von  $d_3$  Aussagen über die Ähnlichkeit der Knotenmarkierungen getroffen werden. Dabei kann leicht ein einfaches Ähnlichkeitsschema auf der Basis der bestehenden HTML-Elementdefinition gegeben werden. HTML-Elemente lassen sich nämlich in zwei auszeichnende Gruppen einteilen [258]:

- *Block-Elemente*: Block-Elemente sind strukturierende HTML-Elemente, wobei sie eigene Absätze im Textfluss markieren. Beispiele: `<ol>`, `<ul>`, `<p>`, `<table>`.

$C_i$	Cluster	Precision	Recall
$C_1$	Mitarbeiter-Webseiten	84%	99%
$C_2$	Lehrveranstaltungsankündigungen	76%	91%
$C_3$	Übersichts-Webseiten verschiedener Themen	65%	92%
$C_4$	Vorlesungs- und Veranstaltungsmaterialien	85%	60%
$C_5$	Downloadseiten für Vorlesungen und Übungen	72%	84%

**Abbildung 5.17:** Bewertung der Clustergüte von  $W_1$ , auf der Basis von Precision und Recall.

- *Inline-Elemente:* Inline-Elemente erzeugen dagegen keine eigenen Absätze im Textfluss. Sie kommen oftmals innerhalb von Block-Elementen vor, wobei sie meistens Text oder wieder Inline-Elemente enthalten. Beispiele: `<a>`, `<basefont>`, `<big>`, `<font>`.

Auf der Grundlage dieser Einteilung kann nun für die in den DOM-Trees vorkommenden HTML-Elemente die Ähnlichkeit untereinander bestimmt werden. Dies geschieht mit Hilfe naheliegender Bewertungsregeln, die Ähnlichkeitswerte anhand der Zugehörigkeit gewisser Elementgruppen vergeben. Die Regeln sind im Wesentlichen:

- Zuweisung eines Ähnlichkeitswertes für Elemente in der gleichen Gruppe, nämlich Block- und Inline-Elemente.
- Minimaler Ähnlichkeitswert für Elemente ungleicher Elementgruppen.
- Zuweisung eines Ähnlichkeitswertes für Elemente in der gleichen Funktionsgruppe. Funktionsgruppen sind hier z.B. `"isHeading"` und `"isTable"`.
- Zuweisung eines Ähnlichkeitswertes für Elemente in unterschiedlichen Funktionsgruppen.

Unter diesen Voraussetzungen kann jetzt unmittelbar das Graphmaß  $d_3$  zur Ähnlichkeitsbestimmung der DOM-Trees verwendet werden. Analog zum Kapitel (5.8.1) wurde die daraus folgende Ähnlichkeitsmatrix als Eingabe für das agglomerative Clusteringverfahren verwendet und zwar auf der Basis von HyGraph [94, 95]. Die Abbildung (5.16) zeigt den entsprechenden Konfigurationsbereich von HyGraph, wobei als Clusterabstand wieder Average Linkage zur Anwendung kam.

Die Abbildungen (5.17), (5.18) präsentieren die Ergebnisse der Ähnlichkeitsmessung mit anschließender Clusterung zweier Websites  $W_1$ <sup>41</sup> und  $W_2$ <sup>42</sup>, wobei die Websites durch die Mengen ihrer DOM-Trees repräsentiert werden. Zur Auswahl

<sup>41</sup><http://www.algo.informatik.tu-darmstadt.de>.

<sup>42</sup><http://www.sec.informatik.tu-darmstadt.de>.

$C_i$	Cluster	Precision	Recall
$C_1$	Mitarbeiter-Webseiten	99%	99%
$C_2$	Übersicht Forschungsthemen	99%	99%
$C_3$	Seminarankündigungen	75%	99%
$C_4$	Lehrveranstaltungsankündigungen	93%	93%
$C_5$	Webseiten für technische Dokumentationen	50%	99%

**Abbildung 5.18:** Bewertung der Clustergüte von  $W_2$ , auf der Basis von Precision und Recall.

einer plausiblen Abbruchstufe hinsichtlich der Clusterlösung wurde die Argumentation des vorherigen Kapitels angewendet. Um einen Eindruck von der Güte der Clusterung zu bekommen, kamen wieder die Maße Recall und Precision zur Anwendung. Falls  $M_r$  die Menge aller relevanten Dokumente und  $M_g$  die Menge der gefundenen Dokumente, bezogen auf ein Cluster  $C$ , bezeichnet, gilt [81]:

$$\begin{aligned} \text{Precision} &:= \frac{|M_r \cap M_g|}{|M_g|}, \\ \text{Recall} &:= \frac{|M_r \cap M_g|}{|M_r|}. \end{aligned}$$

Die Evaluierungsergebnisse zeigen deutlich, dass das eingesetzte Clusteringverfahren auf der Grundlage der berechneten Ähnlichkeitsmatrix Typklassen erzeugte, die strukturell signifikante Webseiten enthalten. Das heißt, die Webseiten einer Klasse besitzen eine auffallend ähnliche Dokumentstruktur und manifestieren damit einen eigenen „Strukturtyp“. Als Beispiel, bezogen auf  $W_1$ , zeigen die Abbildungen (5.19), (5.20) die HTML-Repräsentationen zweier Webseiten aus  $C_2$ . Die hohen Precisionwerte in den Abbildungen (5.17), (5.18) sagen aus, dass die gefundenen Webseiten tatsächlich relevant sind. Dagegen drücken die hohen Recallwerte aus, dass die Webseiten, die für ein Cluster  $C_i$  relevant sind, auch gefunden werden. Bezogen auf Precision fällt dagegen auf, dass das Cluster  $C_5$  in Abbildung (5.18), im Vergleich mit den übrigen Werten, einen niedrigeren Wert besitzt. Weiter besitzt  $C_5$  einen hohen Recallwert. Zusammengefasst bedeutet das: (i) Einige Webseiten, die bezüglich  $C_5$  gefunden wurden, sind nicht relevant und (ii) alle relevanten Webseiten für dieses Cluster wurden gefunden. Bezogen auf Recall ist in Abbildung (5.17) die Situation für  $C_4$  umgekehrt. Insgesamt gesehen sind die hohen Recall- und Precisionwerte als Gütekennzeichen dieser Clusterung positiv zu bewerten.

### 5.8.3 Fazit

Bezogen auf das immer stärker werdende Dokumentaufkommen im World Wide Web sind insbesondere Methoden zur Clusterung strukturell ähnlicher Dokumente und Verfahren zur Informationsextraktion gefordert. Dabei zeigen die

### Ankündigungen:

- Die Ergebnisse der Klausur sind an der Tür S4|03 B110 einzusehen. Klausureinsicht nach Bedarf ab dem 27.8.
- Zum 5. Aufgabenblatt: Je nachdem wie viele Zeilen aus der Datei ldbpmm.txt als gültig erkannt werden, variieren auch die Werte der Parameter. Bei 236 Zeilen ergibt sich z.B.  $183-7+1$ .
- Eine Übungs-Klausuraufgabe steht auf der Seite der Übungen. Die Klausuraufgaben werden einen ähnlichen Stil und Schwierigkeitsgrad haben.
- Die Klausur findet am Montag 21.07.03 um 15:00 im S101/051 statt und dauert 90 Minuten.
- Zum 4. Aufgabenblatt: Die Vorzeichen sollen sowohl beim Menschen als auch bei der Maus alle '+' sein! D.h. Sie müssen die Permutation zuerst in eine ohne Vorzeichen überführen und dann sortieren.
- Zum 4. Aufgabenblatt: Wenn mehrere Gene bei der Maus auf derselben Position liegen, verwenden Sie nur das erste (in menschlicher Reihenfolge) davon.
- Zum 3. Aufgabenblatt: Beachten Sie, dass beim Berechnen der Rückwärtsmatrix auch die Rückwärts-Übergangsw. benutzt werden müssen, d.h. z.B.  $a(l,k)$  statt  $a(k,l)$ . Die Lösung wurde entsprechend angepasst.

### Literatur:

#### Bioinformatik

- Pavel A. Pevzner, "Computational Molecular Biology", MIT Press, 2000.
- Michael S. Waterman, "Introduction to Computational Biology", Chapman & Hall, 1995.
- J.Setubal/J.Meidanis, "Introduction to Computational Molecular Biology", Thompson, 1997.
- Das Skript von Ron Shamir: <http://www.math.tau.ac.il/~rshamir/algmb/01/algmb01.html>
- Das Skript von Volker Heun: <http://www.mayr.informatik.tu-muenchen.de/lehre/2002SS/cb/lecturenotes/>

#### Chemie, Biologie, Genetik

- Rolf Knippers, "Molekulare Genetik", Georg Thieme Verlag, 1997.
- "Biochemie Light", knappe verständliche Einführung in Grundlagen der Biochemie ISBN 3-8171-1638-1
- "Der kleine Alberts" - Lehrbuch der Molekularen Zellbiologie, umfassende Übersicht über moderne Gentechnik, chemische Grundlagen, DNA, RNA, Proteine etc. ISBN 3-527-30493-2

#### Papers

- Kaplan, Shamir, Tarjan, "[Faster and Simpler Algorithm for Sorting Signed Permutations by Reversals](#)"

Abbildung 5.19: HTML-Repräsentation einer Webseite aus  $C_2$ , bezogen auf  $W_1$ .

### Ankündigungen:

- Die Klausureinsicht ist voraussichtlich am Do. 05.8. um 14h (zusammen mit Optimierungsalgorithmen). Der Raum bzw. Terminänderungen stehen dann bei unserer Arbeitsgruppe (E123-E126) an der Tür.
- Die Klausur ist fertig korrigiert. Die Ergebnisse hängen an der Tür vom Raum E126.
- Eine Teillösung der Übungsklausuraufgabe steht auf der Seite der Übungen.
- Die Klausur findet am Montag, den 19. Juli von 9:30h bis 11h im Raum C205 (Piloty-Gebäude) statt.
- Eine Übungsklausuraufgabe steht auf der Seite der Übungen. Die Klausur wird natürlich umfangreicher sein, aber die Art der Aufgaben wird dadurch deutlich.
- Wegen der Verwirrung um die zweite Übung stelle ich noch mal [verschiedene Varianten](#) des Hirschberg-Algorithmus' gegenüber.
- Das Beispiel zum Globalen Alignment gibt es auch [in vollständiger Fassung](#).
- Auf der Materialseite steht ab sofort das Biologie-Skript zur Verfügung.
- Die Vorlesung fällt in der ersten Woche (am 13.4.) leider aus, weil der Vorlesungsraum noch renoviert wird und kein Ersatzraum zur Verfügung steht. Am 20.4. findet die Vorlesung aller Voraussicht nach statt.
- Vorlesungstermin ist Dienstag um 14:25h im Raum C110 S2|02.

### Literatur:

#### Bioinformatik

- Pavel A. Pevzner, "Computational Molecular Biology", MIT Press, 2000.
- Michael S. Waterman, "Introduction to Computational Biology", Chapman & Hall, 1995.
- J.Setubal/J.Meidanis, "Introduction to Computational Molecular Biology", Thompson, 1997.
- Das Skript von Ron Shamir: <http://www.math.tau.ac.il/~rshamir/algmb/01/algmb01.html>

#### Chemie, Biologie, Genetik

- Rolf Knippers, "Molekulare Genetik", Georg Thieme Verlag, 1997.
- "Biochemie Light", knappe verständliche Einführung in Grundlagen der Biochemie, ISBN 3-8171-1638-1
- "Der kleine Alberts" - Lehrbuch der Molekularen Zellbiologie, umfassende Übersicht über moderne Gentechnik, chemische Grundlagen, DNA, RNA, Proteine etc., ISBN 3-527-30493-2

#### Papers

- Kaplan, Shamir, Tarjan, "[Faster and Simpler Algorithm for Sorting Signed Permutations by Reversals](#)"

Abbildung 5.20: HTML-Repräsentation einer weiteren Webseite aus  $C_2$ , bezogen auf  $W_1$ .

Ergebnisse aus Kapitel (5.8.2) wichtige Anwendungen im Bereich der strukturorientierten Filterung web-basierter Dokumente auf. Das bedeutet, dass der Einsatz von  $d_3$  hinsichtlich Massendaten mit dem Ziel zunächst die erforderlichen Ähnlichkeitsmatrizen zu berechnen, sinnvoll ist. Bezogen auf DOM-Strukturen folgt daraus, dass ähnliche Dokumente dazu tendieren, ähnliche Informationen und Layout-Elemente zu besitzen. Weiterhin sagen die Cluster strukturell ähnlicher DOM-Strukturen etwas über ihre Bedeutungen aus. Zum Beispiel hat die Ähnlichkeitsbestimmung und anschließende Clusterung der Webseiten aus den Abbildungen (5.19), (5.20) gezeigt, dass es sich um Webseiten eines Aufzählungstyps handelt. Insgesamt hat man damit Dokumentgruppen gefunden, die auf der Grundlage ihrer Strukturtypen vergleichbar sind.

Anwendungen zur Messung der strukturellen Ähnlichkeit von Website-Strukturen, wobei die Websites in Form von hierarchisierten und gerichteten Graphen repräsentiert sind, treten im Zusammenhang mit Problemstellungen überall dort auf, bei denen (i) die Bildung von Gruppen strukturell ähnlicher Websites gefordert ist und (ii) die spezielle Aufgabenstellung auf Basis der Graphähnlichkeit zu lösen ist. Ein Beispiel für Punkt (ii) wäre z.B. die Bestimmung der strukturellen Auswirkungen eines Veränderungszyklus, bezogen auf eine zeitlich bedingte Folge von Website-Strukturen.

Die hohe Flexibilität hinsichtlich der potenziell zu messenden Strukturaspekte sei als besondere Eigenschaft des neuen Verfahrens nochmals hervorgehoben. Damit kann durch einfache Veränderung der Parametrisierung die Ähnlichkeitsmessung zum einen von Website-Strukturen und zum anderen von DOM-Strukturen fokussiert werden. Abschließend betrachtet wurden damit folgende Ergebnisse im Web Structure Mining erzielt, wobei weitere Anwendungsgebiete in Kapitel (7) thematisiert werden:

- Aufdeckung und bessere Beschreibung bestehender web-basierter Graphstrukturen. Auf Grund der Effizienz ist der Einsatz hinsichtlich Massendaten gegeben.
- Ableitung struktureller Aussagen bezüglich Testkorpora web-basierter Hypertexte, z.B. auf Grundlage von aussagefähigen Verteilungen der Graphähnlichkeitswerte.
- Die Evaluierungsergebnisse der Clusterung aus Kapitel (5.8.2) werden durch hohe Precision- und Recallwerte untermauert. Die in dieser Arbeit entwickelte Methode zur Bestimmung der strukturellen Ähnlichkeit web-basierter Dokumente, leistet neben den bereits erwähnten Anwendungen für Website-Strukturen einen Beitrag im Bereich der strukturellen Dokumentfilterung.

- Abgesehen von den Anwendungen, die aus dem eigentlichen Filterungsprozess resultieren, könnte die strukturorientierte Filterung zukünftig als Vorstufe für eine bessere inhaltsbasierte Kategorisierung betrachtet werden.



# Kapitel 6

## Exkurs: Strukturvorhersage

In Kapitel (5.8.1) wurde auf Basis der Website-Strukturen die Fragestellung untersucht, ob das Graphähnlichkeitsmaß  $d_3$  mit Hilfe eines agglomerativen Clusteringverfahrens in der Lage ist, homogene und aussagekräftige Cluster zu bilden. Da jedoch die Ähnlichkeitswertverteilungen der aus dem WWW extrahierten Website-Strukturen unbekannt sind, kann die Interpretation von Clustering-Experimenten problematisch sein. Daher wird im Folgenden auf der Grundlage bekannter Ähnlichkeitswertverteilungen die über Kapitel (5.8.1) hinausgehende Problemstellung betrachtet, ob mit Hilfe von  $d_3$  strukturelle Beziehungen zwischen vorgegebenen Graphmengen detektiert werden können.

### 6.1 Erkennung struktureller Beziehungen zwischen Graphmengen

Detaillierter betrachtet lässt sich das Problem, welches in diesem Kapitel (6.1) behandelt wird, durch eine wichtige Frage untermauern:

*Im Folgenden sind durch  $C_1$  und  $C_2$  Graphmengen von Website-Strukturen und deren Ähnlichkeitswertverteilungen<sup>1</sup> vorgegeben. Ist auf der Basis von  $d_3$  eine Vorhersage der strukturellen Beziehung zwischen  $C_1$  und  $C_2$  möglich, wobei diese nicht aus den Verteilungen zu erkennen ist?*

Dieses Kapitel drückt die Frage nach der strukturellen Beziehung zwischen den Graphmengen aus. Das bedeutet hier: Wie stark unterscheiden sich die Hauptmasse der Graphen aus  $C_1$  und  $C_2$  strukturell voneinander? Die Konstruktion der Graphmengen wird im Folgenden schrittweise dargestellt:

---

<sup>1</sup>Auf der Basis von  $d_3$ .

1. Konstruktionsvorschrift für  $C_1$ :

- Erzeuge zufällig einen gerichteten Wurzelbaum

$$B_1^{C_1} = (\hat{V}, E_{B_1}^{C_1}), E_{B_1}^{C_1} \subseteq \hat{V} \times \hat{V}$$

unter der Nebenbedingung  $\hat{V} := \left\lceil \frac{|C_1|}{4} \right\rceil, |C_1| > 0$ .

- Ausgehend von  $B_1^{C_1}$  und einem gewählten Startknoten werden jeweils zufällig Across-Kanten, Up-Kanten oder Down-Kanten erzeugt.
- Die nach diesen Schritten konstruierten Website-Strukturen

$$\hat{\mathcal{H}}_m = (\hat{V}, \hat{E}, m_{\hat{V}}, A_{\hat{V}})$$

genügen der Definition (5.3.1), wobei  $A_{\hat{V}} := \{\}$  gilt.

2. Konstruktionsvorschrift für  $C_2$ :

- Erzeuge zufällig eine Folge von gerichteten Wurzelbäumen

$$B_i^{C_2} = (\hat{V}_i, E_{B_i}^{C_2})_{i=1,2,3}, E_{B_i} \subseteq \hat{V}_i \times \hat{V}_i$$

unter den Nebenbedingungen  $\hat{V}_i := \left\lceil \frac{|C_2|}{k_i} \right\rceil, |C_2| > 0, k_1 = 2, k_2 = 10$  und  $k_3 = 20$ .

- Ausgehend von  $B_i^{C_2}$  und einem gewählten Startknoten werden jeweils zufällig Across-Kanten, Up-Kanten oder Down-Kanten erzeugt.
- Basierend auf den Wurzelbäumen  $B_i^{C_2}$  genügen die so konstruierten Website-Strukturen der Definition (5.3.1).

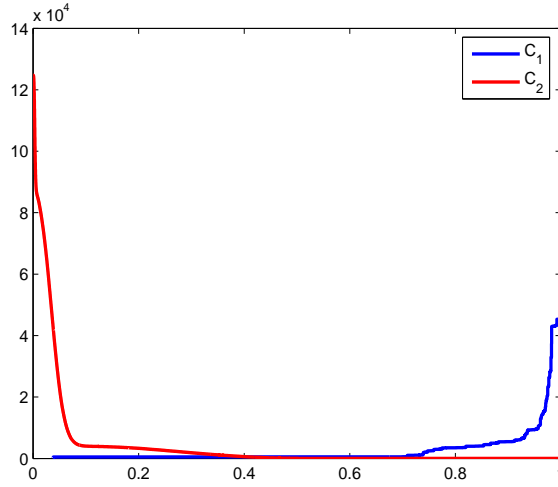
Die Bestimmung der Ähnlichkeitsmatrizen<sup>2</sup>  $(s_{ij})_{ij}, 1 \leq i \leq |C_1|, 1 \leq j \leq |C_1|$ ,  $(s_{ij})_{ij}, 1 \leq i \leq |C_2|, 1 \leq j \leq |C_2|$  und  $(s_{ij})_{ij}, 1 \leq i \leq |C_1 + C_2|, 1 \leq j \leq |C_1 + C_2|$ ,  $s_{ij} \in [0, 1]$  für die Graphmengen  $C_1$  und  $C_2$  wurde auf Basis der Parameterbelegung gemäß Punkt (5) der Definition (5.8.1) durchgeführt.

Das eigentliche Experiment kann auf der Grundlage dieser Voraussetzungen folgendermaßen unterteilt werden:

1. Graphische Darstellung der Ähnlichkeitswertverteilungen mit Hilfe der Matrizen  $(s_{ij})_{ij}, 1 \leq i \leq |C_1|, 1 \leq j \leq |C_1|$  und  $(s_{ij})_{ij}, 1 \leq i \leq |C_2|, 1 \leq j \leq |C_2|$ .

---

<sup>2</sup>Auf der Basis von  $s = d_3$ .



**Abbildung 6.1:** Das Schaubild zeigt die gerankten Ähnlichkeitswerte der Graphmengen  $C_1$  und  $C_2$ . Es gilt  $\theta = 0.5$  und  $|C_1| = |C_2| = 500$ . X-Achse: Ähnlichkeitswert  $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \in [0, 1]$ . Y-Achse: Anzahl der Graphpaare.

2. Bilde die Menge  $C := C_1 \cup C_2$  und berechne die Ähnlichkeitsmatrix  $(s_{ij})_{ij}$ ,  $1 \leq i \leq |C_1 + C_2|$ ,  $1 \leq j \leq |C_1 + C_2|$ .
3. Durch die Mischung der Graphmengen  $C_1$  und  $C_2$  sind während der Berechnung der Ähnlichkeitsmatrix neue Graphpaarungen entstanden, wobei hier die Gesamtheit der Paare durch  $(\hat{\mathcal{H}}_m^i, \hat{\mathcal{H}}_m^j)_{ij}$  bezeichnet wird. Konstruiere die Mengen  $C_1^{\text{new}}$  und  $C_2^{\text{new}}$  nach folgendem Kriterium: Gilt die Ungleichung

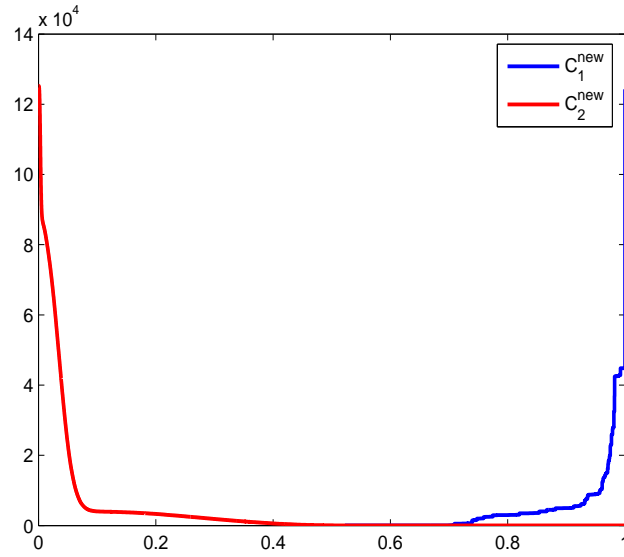
$$d_3((\hat{\mathcal{H}}_m^i, \hat{\mathcal{H}}_m^j)) \geq \theta, \forall (\hat{\mathcal{H}}_m^i, \hat{\mathcal{H}}_m^j) \in (\hat{\mathcal{H}}_m^i, \hat{\mathcal{H}}_m^j)_{ij},$$

dann ordne dieses Paar der Menge  $C_1^{\text{new}}$ , ansonsten der Menge  $C_2^{\text{new}}$  zu. Dabei bezeichnet  $\theta$  den Schwellwert der Mengenkonstruktion, wobei sich  $\theta$  über den Verteilungsschnittpunkt<sup>3</sup> berechnet.

4. Bestimme nun die Anzahl der Graphpaarungen, die sich mit Graphen aus  $C_1$  und  $C_1^{\text{new}}$  bildeten. Bezogen auf  $C_2$  und  $C_2^{\text{new}}$  ist das Vorgehen analog. Die daraus resultierenden Paarungshäufigkeiten, die mit der Kardinalität  $|C_1| + |C_2|$  normiert werden, gelten als Vorkommenswahrscheinlichkeiten für die entsprechenden Graphen in  $C_1^{\text{new}}$  bzw.  $C_2^{\text{new}}$ .
5. Die Graphen, die auf der Basis dieser Zählung die größte Häufigkeit bezüglich  $C_1^{\text{new}}$  und  $C_2^{\text{new}}$  erzielen, werden eindeutig der entsprechenden Menge zugeordnet.

---

<sup>3</sup>Dieser wurde in Abbildung (6.1) zu  $\theta = 0.5$  berechnet.



**Abbildung 6.2:** Das Schaubild zeigt die gerankten Ähnlichkeitswerte von  $C_1^{\text{new}}$  und  $C_2^{\text{new}}$ . Die Auftrennung der gemischten Menge  $C = C_1 \cup C_2$  erfolgte mit  $\theta = 0.5$ . X-Achse: Ähnlichkeitswert  $d_3(\hat{\mathcal{H}}_m^1, \hat{\mathcal{H}}_m^2) \in [0, 1]$ . Y-Achse: Anzahl der Graphpaare.

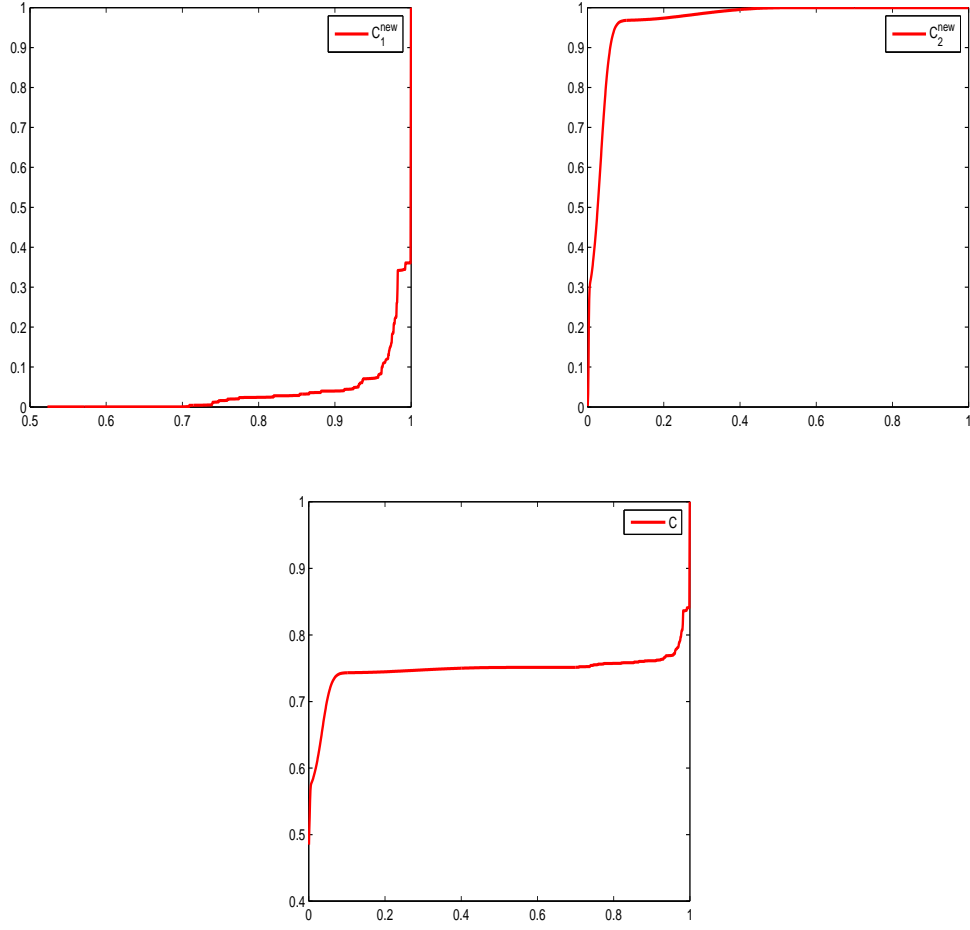
Auf dieser Konstruktion basierend kann die Kernfrage der Untersuchung damit beantwortet werden, wie gut die Auftrennung der gemischten Graphmenge mit Hilfe von  $d_3$  gelingt.

## 6.2 Ergebnisse

Die Interpretation der Abbildung (6.1) ergibt, dass die Hauptmasse der Graphen aus  $C_1$  untereinander sehr ähnlich<sup>4</sup> sind. Bezogen auf die Gesamtanzahl der Graphpaare aus  $C_1$  existieren nur wenige Paarungen, die einen Ähnlichkeitswert  $d_3 \leq 0.8$  besitzen. Hinsichtlich  $C_2$  ist die Situation gerade umgekehrt: Der Hauptanteil der Graphen aus  $C_2$  ist untereinander extrem unähnlich<sup>5</sup>. Die Abbildung (6.2) zeigt die Schaubilder der gerankten Ähnlichkeitswerte von  $C_1^{\text{new}}$  und  $C_2^{\text{new}}$ , wobei die Kurvenverläufe identisch mit denen aus Abbildung (6.1) erscheinen. Eine endgültige Aussage über die strukturelle Beziehung zwischen den Graphmengen kann aber nur eine Auftrennung der Graphmengen ergeben, da die Ähnlichkeitswertverteilungen keine Rückschlüsse über die Beziehung zwischen den Graphklassen erlauben. Um weiter eine Vorstellung über die kumulativen Ähnlichkeitswertverteilungen von  $C_1^{\text{new}}$  und  $C_2^{\text{new}}$  zu bekommen, betrachte

<sup>4</sup>Auf der Basis von  $d_3$ .

<sup>5</sup>Auf der Basis von  $d_3$ .



**Abbildung 6.3:** Kumulative Ähnlichkeitswertverteilungen von  $C_1^{\text{new}}$ ,  $C_2^{\text{new}}$  und  $C$ . Auf der  $X$ -Achse ist jeweils der Ähnlichkeitswert  $d_3 \in [0, 1]$  aufgetragen. Die  $Y$ -Achse bezeichnet den Prozentsatz der Graphen, die einen Ähnlichkeitswert  $d_3 \leq X$ -Wert besitzen.

man die Abbildung (6.3). Zum Beispiel besitzen 90% der Graphen aus  $C_2^{\text{new}}$  einen Ähnlichkeitswert  $d_3 \leq 0.1$ . Somit gelten diese Graphen auf der Basis von  $d_3$  als sehr unähnlich. Dagegen gilt für  $C_1^{\text{new}}$ , dass bereits 20% der Graphen die Ungleichung  $d_3 \leq 0.95$  erfüllen. Weiterhin zeigt die Abbildung (6.3) die kumulative Ähnlichkeitswertverteilung der gemischten Graphmenge  $C$ .

Die bereits geäußerte Vermutung, dass sich die Hauptmasse der Graphen aus  $C_1$  und  $C_2$  stark voneinander unterscheiden, wird nun auf die Kernfrage zurückgeführt, wie ausgeprägt die Trennung der gemischten Graphmenge  $C$  ausfällt. Mit anderen Worten muss damit die Frage beantwortet werden, wieviel Prozent der Graphen der Ausgangsmenge  $C_1$  bzw.  $C_2$  gemäß Punkt (4), (5) der Menge  $C_1^{\text{new}}$  bzw.  $C_2^{\text{new}}$  zugeordnet wurden. Das optimale Ergebnis der Mengentrennung

wäre, dass die neu gewonnenen Graphmengen  $C_1^{\text{new}}$  und  $C_2^{\text{new}}$  identisch mit den Ursprungsmengen  $C_1$  und  $C_2$  sind.

Unter der Voraussetzung  $|C_1| = |C_2| = 500$  gilt für die Kardinalität der gemischten Menge offensichtlich  $|C| = 1000$ . Um das Ergebnis der Trennung zu interpretieren, wurden zunächst die oben genannten Prozentzahlen berechnet. Die abschließende Auswertung ergab, dass sich 99.8% der ursprünglichen Graphen von  $C_1$  in  $C_1^{\text{new}}$  befanden. Das bedeutet, dass lediglich ein Graph, der ursprünglich  $C_1$  angehörte, nun  $C_2^{\text{new}}$  zugeordnet wurde. Dagegen wurden die Ursprungsgraphen aus  $C_2$  zu 100% richtig der neuen Menge  $C_2^{\text{new}}$  zugeordnet.

## 6.3 Fazit

Die Tatsache, dass die Hauptmasse der Graphen aus  $C_1$  bzw.  $C_2$  untereinander strukturell unähnlich bzw. ähnlich ist, lässt keine endgültige Aussage über die Beziehung zwischen  $C_1$  und  $C_2$  zu. Deshalb bestätigen die Ergebnisse die Hypothese auf der Basis der Auftrennung, dass sich der Hauptanteil der Graphen aus  $C_1$  und  $C_2$  stark voneinander unterscheiden. In diesem Fall kommt die ausgeprägte strukturelle Verschiedenheit dieser Graphklassen durch deren Konstruktion zustande.

Diese Auswertung zeigt deutlich, dass das Graphähnlichkeitsmaß  $d_3$  in der Lage ist, komplexe Graphstrukturen in Form von hierarchisierten und gerichteten Graphen strukturell zu erkennen und im Sinne der Trennung zu klassifizieren. Die Lösung der hier behandelten Problemstellung war nicht aus der Betrachtung der Ähnlichkeitswertverteilungen abzuleiten. Als weitergehende Anwendung ist die binäre Klassifikation von Graphmengen hierarchisierter und gerichteter Graphen von großem Interesse. Damit kann die Frage der Gleichheit von zwei spezifischen Graphmengen beantwortet werden.

# Kapitel 7

## Zusammenfassung und Ausblick

Kapitel (7) fasst die Ergebnisse dieser Arbeit zusammen. Darüber hinaus werden in Form eines Ausblicks Bereiche angegeben, in denen das Hauptergebnis der Arbeit, das Graphähnlichkeitsmodell für hierarchisierte und gerichtete Graphen, über die Kapitel (5.8.1), (5.8.2), (6) hinaus zukünftig Anwendung finden kann. Abschließend werden weiterführende Fragestellungen und Ansatzpunkte für zukünftige Untersuchungen aufgezeigt.

### 7.1 Zusammenfassung der Ergebnisse

In der vorliegenden Arbeit wurden strukturelle Aspekte web-basierter Hypertexte untersucht. Dabei stand die Entwicklung solcher graphentheoretischer Analysemethoden im Vordergrund, die anwendungsorientierte Problemstellungen im Web Structure Mining lösen. Die Untersuchungen dieser Arbeit lassen sich in zwei aufeinander aufbauende Teile untergliedern:

- Kapitel (3) zeigt die Grenzen der inhaltsbasierten Kategorisierung web-basierter Einheiten auf. Als Basis von Kapitel (3) gilt eine grundlegende Arbeit [156], die die Phänomene Polymorphie und funktionale Äquivalenz definiert. Weiterhin thematisiert [156] die aus der Polymorphie resultierenden Probleme bei der Kategorisierung hypertextueller Einheiten. In Kapitel (3.4) wurde dazu ein Experiment entworfen, welches die inhaltsbasierte Kategorisierung englischsprachiger Konferenz/Workshop-Websites im Bereich Mathematik und Informatik fokussiert. Die experimentellen Ergebnisse [61, 156] aus Kapitel (3.6) zeigen, dass die Performance-Evaluierung der SVM-Kategorisierung zu niedrigen Precision- und hohen Recallwerten führte. Die sehr niedrige Trennschärfe der Kategorien weist auf eine extreme Mehrfachkategorisierung hin, das heißt die zu kategorisierenden Websei-

ten wurden in den meisten Fällen mehreren Kategorien zugeordnet. Diese Ergebnisse untermauern nachhaltig die Hypothese, dass Polymorphie und funktionale Äquivalenz charakteristische Eigenschaften web-basierter Einheiten sind. Da das Experiment aus Kapitel (3.4) im Rahmen des bekannten Vektorraummodells erfolgte, wurde in Kapitel (5) die Entwicklung eines neuen Ansatzes zur Modellierung multimedialer Dokumentstrukturen dargestellt.

- Auf der Suche nach einem adäquaten Ansatz zur Modellierung web-basierter Hypertexte wurde in Kapitel (5) ein Verfahren vorgestellt, welches die graphbasierte Struktur der Hypertexte ganzheitlich berücksichtigt. Zusammen mit den Ergebnissen aus Kapitel (3) weist dieses Verfahren nach: Graphentheoretische Analysemethoden im Hinblick auf das Web Mining sind nur dann sinnvoll, wenn sie aussagekräftige und effiziente *strukturelle* Vergleiche graphbasierter Hypertexte ermöglichen. Damit ist der wesentliche Beitrag dieser Arbeit ein graphentheoretisches Modell, welches die Bestimmung der strukturellen Ähnlichkeit von Hypertextstrukturen in Form hierarchisierter und gerichteter Graphen beschreibt. Kurz gefasst lassen sich die Entwicklungs- und Modellierungsschritte wie folgt darstellen:
  - Es wurden strukturelle Parameter gesucht, die einerseits effizient berechenbar und andererseits aussagekräftig sind. Erste Zwischenergebnisse legen die Verwendung von Gradsequenzvektoren gerichteter Graphen nahe, welche aus den Adjanzenzmatrizen einfach zu berechnen sind. Allerdings sagen Vergleiche solcher Gradsequenzen wenig über die gemeinsame Graphstruktur aus (Kapitel (5.1), (5.2)).
  - Der entscheidende Schritt bestand darin, die Knotensequenzen und die dadurch induzierten Gradsequenzen ebenenweise zu betrachten. Die Frage nach der strukturellen Ähnlichkeit zweier graphbasierter Hypertexte in Form hierarchisierter und gerichteter Graphen, wurde dadurch gleichbedeutend mit der Bestimmung eines optimalen Sequenz-Alignments der zu Grunde liegenden Grundsequenzen (Kapitel (5.4), (5.5)).
  - Die Graphähnlichkeitsmaße, die auf dieser Grundidee und den definierten Bewertungsfunktionen beruhen, besitzen die Eigenschaften von Backward-Ähnlichkeitsmaßen. Auf Grund der Konstruktion können durch einfache Parameteränderungen hinsichtlich der Bewertungsfunktionen vielfältige strukturelle Teilaspekte während der Ähnlichkeitsbestimmung berücksichtigt werden (Kapitel (5.6), (5.8.1), (5.8.2)).

Dieses Modell bildet den Schlüssel für vielfältige Anwendungen, die über die Problemstellungen des Web Structure Mining hinausgehen. Um das Graphähnlich-



keitsmodell aus Kapitel (5) zu bewerten, wurden in den Kapiteln (5.8.1), (5.8.2) experimentelle Untersuchungen durchgeführt. Insgesamt wurden damit qualitativ wichtige Ergebnisse im Web Structure Mining erzielt:

- Auf der Basis der ähnlichkeitsbasierten Untersuchungsmethode wurden die graphentheoretischen Beschreibungsmöglichkeiten bestehender Hypertexte deutlich verbessert. Auf Grund des neuen Graphähnlichkeitsmodells können strukturelle Aussagen bezüglich Testkorpora web-basierter Einheiten mit Hilfe der Ähnlichkeitswertverteilungen leicht getroffen werden.
- Anstatt der Verwendung von graphentheoretischen Indizes [29, 63, 155], die strukturelle Ausprägungen auf eine Maßzahl abbilden, können nun auf der Grundlage der Ähnlichkeitsmatrizen multivariate Analyseverfahren verwendet werden [63]. Als wichtiger Vertreter wurden hier die Clusteringverfahren<sup>1</sup> ausgewählt. Da in dieser Arbeit Ähnlichkeitsmatrizen die Eingabe der Clusteringverfahren bilden, sind alle Strukturmerkmale der komplexen graphbasierten Hypertexte in der Ähnlichkeitsmatrix abgebildet. Die Anwendungsmöglichkeit der Clusteringverfahren auf die berechneten Ähnlichkeitsmatrizen stellt eine wesentliche Verbesserung im Vergleich zu den Analysemöglichkeiten auf Grundlage graphentheoretischer Indizes dar, da es sich um Struktur entdeckende Verfahren handelt, die mehrere Objekteigenschaften gleichzeitig berücksichtigen.
- Die Evaluierungsergebnisse der Cluster-Gütebestimmung web-basierter Dokumente in Form von DOM-Strukturen weisen im Wesentlichen hohe Precision- und Recallwerte auf. Damit ist ein sinnvoller Einsatz in der strukturorientierten Filterung multimedialer Dokumentstrukturen gegeben.
- Überall dort, wo sich strukturorientierte und ähnlichkeitsbasierte Problemstellungen web-basierter Dokumente ergeben, kann das neue Graphähnlichkeitsmodell eingesetzt werden. Durch einfache Parameteränderung kann zum einen die strukturelle Ähnlichkeit von Website-Strukturen und zum anderen von DOM-Strukturen bestimmt werden. Während Website-Strukturen in Form von unmarkierten hierarchisierten und gerichteten Graphen repräsentiert werden, stellen DOM-Strukturen knotenmarkierte gerichtete Wurzelbäume dar.

Zusammenfassend formuliert belegen die experimentellen Ergebnisse der Kapitel (5.8.1), (5.8.2) die These, dass die graphbasierte Repräsentation hypertextueller Dokumente einen zentralen Ausgangspunkt für die Modellierung und ähnlichkeitsbasierte Analyse web-basierter Dokumente darstellt. Insbesondere zeigen die Ergebnisse aus Kapitel (6), dass das in dieser Arbeit für Evaluierungen

---

<sup>1</sup>Siehe Kapitel (2.4).

eingesetzte Graphähnlichkeitsmaß<sup>2</sup> zur Erkennung und Klassifikation komplexer Graphstrukturen geeignet ist.

Abgesehen von den Anwendungen im Web Structure Mining, wurde das Graphähnlichkeitsmodell aus Kapitel (5) bereits erfolgreich zur Klassifikation großer<sup>3</sup> ungerichteter Graphen eingesetzt [73]. Ein wichtiger Anwendungsbereich ist dabei ist die Unterscheidung von Tumorstadien. Die Klassifikationsmethode und der Anwendungshintergrund werden am Ende des Kapitels (7.2) detaillierter erklärt.

## 7.2 Ausblick

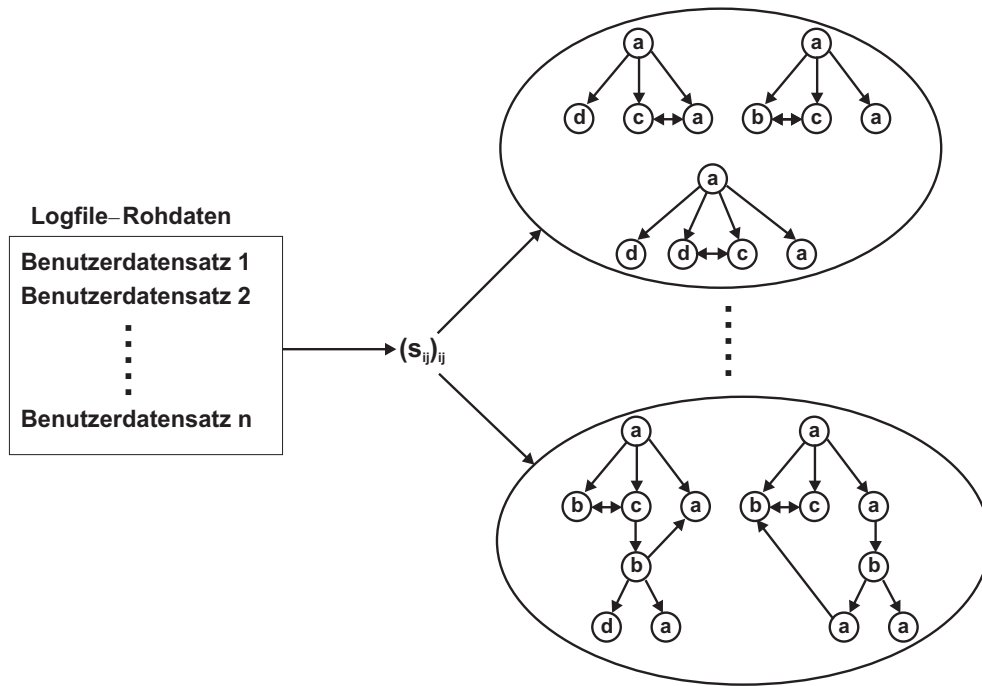
Das Web Mining ist im Vergleich zum klassischen Information Retrieval ein junges Forschungsgebiet. In Kapitel (2.2.2) wurden die Kernbereiche des Web Mining vorgestellt, wobei es insgesamt die Teilbereiche Web Structure Mining, Web Usage Mining und Web Content Mining umfasst. Da die in dieser Arbeit erzielten Ergebnisse besonders dem Web Structure Mining zuzuordnen sind, werden im Folgenden die Ergebnisse in den Rahmen der übrigen Teilbereiche des Web Mining gestellt.

Das Web Usage Mining besitzt die Aufgabe, anhand spezifischer Muster das Benutzerverhalten zu analysieren, wobei die Analysemuster mit Hilfe von Web-Logs gewonnen werden. In der Praxis findet das Web Usage Mining breite Anwendung, z.B. wird es zur Untersuchung des Kaufverhaltens und zur Qualitätsanalyse von Websites eingesetzt. Ein mögliches Anwendungsszenario der Ergebnisse aus Kapitel (5) im Web Usage Mining wird nun wie folgt beschrieben: Die vollständigen Benutzerdaten werden auf der Basis von Log-Dateien zunächst in knotenmarkierte, hierarchisierte und gerichtete Graphen transformiert. Da in den meisten Fällen eine eindeutige Einstiegs-Webseite vorliegt, ist die Berechnung der Kantenstruktur solcher Graphen leicht möglich. Die Knotenmarkierungen können beispielsweise durch unterschiedliche Dokumenttypen repräsentiert werden. Damit sind die Voraussetzungen zur Anwendung des Analyseansatzes aus den Kapiteln (5.8.1), (5.8.2) gegeben. Unter der Betonung gewünschter Strukturaspekte während der Berechnung der Ähnlichkeitsmatrix kann das agglomerative Clusteringverfahren angewendet werden. Dabei ist auch die Anwendung weiterer Clusteringverfahren aus Kapitel (2.4) möglich. Entscheidend für alle Clusteringverfahren ist dabei, dass die Cluster auf der Basis einer anwendungsorientierten Problemstellung interpretiert und damit nicht isoliert betrachtet werden. Abbildung (7.1) zeigt schematisch die Übertragung des in dieser Arbeit verwendeten ähnlichkeitsbasierten Analyseansatzes auf das Web Usage Mining.

---

<sup>2</sup> $d_3$  aus Satz (5.6.4).

<sup>3</sup>Falls  $G = (V, E)$  einen ungerichteten Graph bezeichnet, so gilt  $|V| \approx 10^5$ .



**Abbildung 7.1:** Schematische Darstellung der ähnlichkeitsbasierten Graphanalyse im Web Usage Mining.

Als Ergebnis entstehen Cluster, in denen auf der Basis des verwendeten Graphähnlichkeitsmaßes strukturell ähnliche Graphen zusammengeschlossen werden. Die Cluster stellen dabei Benutzergruppen dar, wobei Benutzer, die sich durch ähnliches Navigationsverhalten auszeichnen, Cluster mit ähnlichen Graphmustern erzeugen. Das beschriebene Anwendungsszenario ist somit leicht im *E-Learning* [160] anwendbar, falls das Ziel darin besteht, graphbasierte Benutzergruppen zu identifizieren. Da die Benutzerdaten bei Evaluierungen von Lernplattformen [13] aufgezeichnet werden, können diese Daten auf der Grundlage der vorgestellten Analysemethoden wie beschrieben untersucht und interpretiert werden. Weiterhin besitzen die gefundenen Gruppen eine lernpsychologische Bedeutung, da sich in den Graphmustern z.B. benutzerspezifische Präferenzen und konditionales Vorwissen widerspiegeln.

Ein weiteres zukünftiges Anwendungsgebiet ist das Web Content Mining [139], welches die Informationsextraktion hinsichtlich web-basierter Dokumente zum Ziel hat. Dabei tritt jedoch häufig das Problem auf, dass ohne ausreichend vorhandenes Wissen bezüglich der Dokumentstruktur, zu wenig über den Dokumentinhalt ausgesagt werden kann. Dieser negative Aspekt stellt ein Problem dar, falls das weiterführende Ziel die inhaltsbasierte Kategorisierung web-basierter Dokumente auf der Basis verschiedener Themengebiete ist. Damit ist es möglich, Verfahren, die in erster Linie zur inhaltsbasierten Klassifikation geeignet sind, mit

den Modellierungsmöglichkeiten der in dieser Arbeit vorgestellten graph- und ähnlichkeitsbasierten Analyse zu kombinieren.

Ein Anwendungsbereich der ähnlichkeitsbasierten Graphanalyse, der von den bisher genannten abweicht, stellt die Klassifikation großer ungerichteter Graphen dar. Das Graphähnlichkeitsmodell aus Kapitel (5) wurde bereits erfolgreich in diesem Gebiet angewendet [73], wobei einerseits künstlich generierte und andererseits Daten aus *Microarray*-Experimenten [42] zur Verwendung kamen. Die Microarray-Technologie hat besonders in jüngster Zeit eine revolutionäre Entwicklung erfahren, da sie zu großen Fortschritten in der medizinisch-klinischen Forschung geführt hat. Ein auszeichnender Faktor ist dabei, dass die jeweiligen Experimente innerhalb kürzester Zeit durchgeführt werden können. In der *Gentechnik* sind die Microarray-Experimente ebenfalls populär geworden. Die bekannten DNA-Microarrays ordnen mehrere tausend DNA-Sequenzen [145] reihenweise an, wobei sie oft zur Analyse unbekannter DNA-Sequenzen eingesetzt werden. Zurückkommend auf die Klassifikation großer Graphen kann der Analyseansatz auf der Basis der in dieser Arbeit entwickelten Graphähnlichkeitsmaße zusammengefasst dargestellt werden: Auf der Grundlage eines in [73] entwickelten Verfahrens, welches große Graphen eindeutig in eine Menge  $H$  hierarchisierter und gerichteter Graphen zerlegt, wurde eine binäre Graphklassifikation definiert. Das heißt, die Frage nach der Ähnlichkeit zweier großer Graphen  $G_1$  und  $G_2$  kann auf Grundlage der Ähnlichkeitswertverteilungen der zugehörigen Mengen  $H_1$  und  $H_2$  beantwortet werden. Die binäre Klassifikation ist also im folgenden Sinn definiert:

*$G_1$  und  $G_2$  sind genau dann ähnlich, wenn die zugehörigen Ähnlichkeitswertverteilungen der Mengen  $H_1$  und  $H_2$  ähnlich sind.*

Mit dieser Konstruktion kann nun die Frage beantwortet werden, ob die Ursprungsgraphen  $G_1$  und  $G_2$  derselben Graphklasse angehören. Die in [73] durchgeführten Experimente zur Graphklassifikation wurden zum einen mit Small World-Graphen [242] und zum anderen mit Random-Graphen [25] durchgeführt. Die erzielten Ergebnisse stellen eine wesentliche Verbesserung bezogen auf die bisherigen Ansätze im Vergleich großer Graphen dar. Da bestehende Ansätze [169, 172] oft auf Berechnungen der Gradverteilungen oder Graphzusammenhangsrelationen beruhen, geben die so gewonnenen Verteilungen zu wenig von der eigentlichen Graphstruktur wieder. Die Evaluierung in [73] zeigte, dass ein Vergleich der Ähnlichkeitswertverteilungen wesentlich aussagekräftiger ist. In einem zweiten Schritt wurde die eben beschriebene Klassifikationsmethode auf die bereits erwähnten Microarray-Daten aus Gebärmutterhalskrebs-Experimenten angewendet, mit dem Ziel, Tumorstadien zu unterscheiden. Die Ergebnisse dieser Untersuchung [73] wurden unter der folgenden Voraussetzung interpretiert: Falls

das klinische Stadium des erkrankten Gewebes durch ungerichtete und unmarkierte Graphen ausdrückbar ist, so konnte auf der Basis der Ähnlichkeitswertverteilungen eine deutliche Unterscheidung und damit unterschiedliche Krankheitsstadien festgestellt werden. In der Zukunft sind weitere Experimente geplant, mit dem Hauptziel, die Tumorerkennung in besonders frühen Stadien zu verbessern.

## 7.3 Weiterführende Fragestellungen

Die weiterführenden Fragestellungen, die sich im Laufe der vorliegenden Arbeit herausbildeten, werden folgendermaßen charakterisiert:

- In Kapitel (5.4) wurde die Transformation der Graphen in formale Knotensequenzen betrachtet. Dabei ist die weitere Konstruktion des Graphähnlichkeitsmodells darauf aufgebaut, die strukturellen Vergleiche auf der Basis der Alignments der induzierten Eingangsgrad- und Ausgangsgradsequenzen ebenenweise durchzuführen. An dieser Stelle wären weitere Untersuchungen sinnvoll, mit dem Schwerpunkt, hierarchisierte und gerichtete Graphen in formale Zeichenketten abzubilden. Dabei sind insbesondere solche Transformationen gesucht, die einerseits die Kantenordnung der Graphen so ausgeprägt wie möglich erhalten und andererseits nicht zwangsgläufig in der ebenenorientierten Betrachtungsweise münden. Beispielsweise beschreiben ROBLES-KELLY et al. [182] ein Verfahren, das mit Hilfe spektraler<sup>4</sup> Methoden beliebige Graphen in formale Zeichenketten transformiert. Der nächste Schritt wäre wieder die Definition entsprechender Bewertungsfunktionen, um die Ähnlichkeit der in Sequenzen transformierten Graphen zu bestimmen. Solche Transformationsverfahren wären dann auf der Basis experimenteller Auswertungen zu evaluieren.
- Das in dieser Arbeit vorgestellte Graphähnlichkeitsmodell besitzt eine auszeichnende Grundeigenschaft: Die Gesamtheit der auf den Graphenebenen abgeleiteten Ähnlichkeitswerte bilden nicht per se das gewünschte Graphähnlichkeitsmaß, sondern auf der Grundlage der Ähnlichkeitswerte kann jederzeit ein neues Maß definiert werden. In der Zukunft wäre die Erprobung und Evaluierung neu definierter Maße sinnvoll, mit dem Ziel, die Güte des Strukturerkennungsverhaltens zu bestimmen. Diese Untersuchungen sollten sich zum einen auf web-basierte Hypertexte und zum anderen auf die beschriebene Tumorunterscheidung konzentrieren.
- Soll die Anwendung der Ähnlichkeitsmaße auf der Basis neuer Testkorpora, bezogen auf unterschiedliche Graphähnlichkeitsprobleme erfolgen, so wären

---

<sup>4</sup>Siehe Kapitel (4.1.1).

ausführliche Parameterstudien nötig. Dabei ist aber zu beachten, dass keine universellen Parametersätze für die Ähnlichkeitsmessung unterschiedlicher Graphklassen existieren, siehe z.B. [73].

Mit dieser Dissertation hoffe ich, einen Beitrag geleistet zu haben, um anwendungsorientierte Problemstellungen im Web Structure Mining zufriedenstellender als bisher zu lösen. Da die Bestimmung der strukturellen Ähnlichkeit von Graphen in vielen Forschungsbereichen immer noch ein sehr herausforderndes Problem darstellt, wünsche ich mir, dass die darauf bezogenen Ergebnisse dieser Arbeit einen positiven Ausgangspunkt für weiterführende Arbeiten bilden.

# Literaturverzeichnis

- [1] Adamic L., Huberman B.: *Power-law distribution of the world wide web*, Science, Vol. 287 (2115a), 2000
- [2] Altschul S. F., Madden T. L., Schaffer A. A., Zhang J., Zhang Z., Miller W., Lipman D. J.: *Gapped BLAST and PSI-BLAST: A new generation of protein database search programs*, Nucleic Acids Res., Vol. 25, 1997
- [3] Anderberg M. R.: *Cluster Analysis for Applications. Probability and Mathematical Statistics*, Academic Press, New York, Vol. 19, 1973
- [4] Arabie P., Lawrence J. H., De Soete G.: *Clustering and Classification*, World Scientific Publishers, 1996
- [5] Arvind V., Kurur P. P.: *Graph isomorphism is in SPP*, 43-th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2002, 743-752
- [6] Backhaus K., Erichson B., Plinke W., Weiber R.: *Multivariate Analysemethoden*, Springer Verlag, 2003
- [7] Baeza-Yates R., Ribeiro-Neto B.: *Modern Information Retrieval*, ACM Press, Addison Wesley, 1999
- [8] Bang-Jensen J., Gutin G.: *Digraphs. Theory, Algorithms and Applications*, Springer Verlag, London-Berlin-Heidelberg, 2002
- [9] Bar-Hillel Y.: *Language and Information. Selected Essays on their Theory and Application*, Addison-Wesley Series in Logic, Addison-Wesley Publishing, 1964
- [10] Basak S. C., Nikoli S., Trinajsti N., Amic D., Beslo D.: *QSPR modeling: Graph connectivity indices versus line graph connectivity indices*, Journal of Chemical Information and Computer Sciences, Vol. 40, 2000, 927-933
- [11] Batagelj V.: *Similarity measures between structured objects*, MATH/CHEM/COMP. Proceedings of an International Course and Conference on the Interfaces between Mathematics, Chemistry and Computer Sciences, Dubrovnik/Yugoslavia, 1988

- [12] Bavelas A.: *Communication patterns in task-oriented groups*, J. Acoust. Soc. Amer., 1950, 725-730
- [13] Baumgartner P., Häfele H., Maier-Häfele K.: *E-Learning Praxishandbuch. Auswahl von Lernplattformen*, Studien Verlag, 2002
- [14] Behzad M., Chartrand G., Lesniak-Foster L.: *Graphs & Digraphs*, Prindle, Weber & Schmidt International Series, 1979
- [15] Bellman R.: *Dynamic Programming*, Princeton University Press, 1957
- [16] Bellman R.: *Dynamische Programmierung und selbstanpassende Regelprozesse*, Oldenbourg Verlag, 1967
- [17] Berge C.: *Graphs and Hypergraphs*, American Elsevier Pub., 1973
- [18] Berkhin P.: *Survey of clustering data mining techniques*, Technical Report, Accrue Software, 2002
- [19] Bernes-Lee T.: *Information management: A proposal*, <http://www.w3.org/History/1989/proposal.html>
- [20] Bernes-Lee T.: *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web*, HarperBusiness Publishing, 2000
- [21] Berry M. J. A., Linoff G.: *Data Mining Techniques for Marketing, Sales and Customer Support*, Wiley, New York, 1997
- [22] Berthold M., Hand D. J.: *Intelligent Data Analysis*, Springer Verlag, 2003
- [23] Bock H. H.: *Automatische Klassifikation. Theoretische und praktische Methoden zur Gruppierung und Strukturierung von Daten*, Studia Mathematica - Mathematische Lehrbücher, Vandenhoeck & Ruprecht Verlag, 1974
- [24] Boppana R., Hastad J., Zachos S.: *Does co-NP have short interactive proofs?*, Information Processing Letters, Vol. 25, 1987, 127-132
- [25] Bollabás B.: *Modern Graph Theory*, Graduate Texts in Mathematics, Springer Verlag, New York, 1998
- [26] Bornholdt S.: *Handbooks of Graphs and Networks*, Wiley VCH, 2002
- [27] Borgelt C., Berthold M. R.: *Mining molecular fragments: Finding relevant substructures of molecules*, Proceedings of the 2002 International Conference on Data Mining, 2002, 51-58
- [28] Botafogo R. A., Shneiderman B.: *Identifying aggregates in hypertext structures*, Proc. ACM Conf. of Hypertext, 1992, 38-47



- [29] Botafogo R. A., Rivlin E., Shneiderman B.: *Structural analysis of hypertexts: Identifying hierarchies and useful metrics*, ACM Trans. Inf. Syst. 10 (2), 1992, 142-180
- [30] Botafogo R. A.: *Cluster analysis for hypertext systems*, SIGIR 1993, 116-125
- [31] Brin S., Page L.: *The anatomy of a large-scale hypertextual web search engine*, Proc. of the 7-th WWW Conf., 1998
- [32] Broder A., Kumar R., Maghoul F., Raghavan P., Rajagopalan S., Stata R., Tomkins A., Wiener J.: *Graph structure in the web: Experiments and models*, Proc. of the 9-th WWW Conf., Amsterdam, 2000
- [33] Bronstein I. A., Semendjajew A., Musiol G., Mühlig H.: *Taschenbuch der Mathematik*, Harri Deutsch Verlag, 1993
- [34] Bunke H.: *Attributed programmed graph grammars and their application to schematic diagram interpretation*, IEEE Trans. Pattern Anal. Mach. Intell., Vol. 4, 1982, 574-582
- [35] Bunke H.: *What is the distance between graphs?*, Bulletin of the EATCS, Vol. 20, 1983, 35-39
- [36] Bunke H., Allermann G.: *A metric on graphs for structural pattern recognition*, In: SIGNAL PROCESSING II: Theorie and Applications. Editor: H. W. Schüssler, 1983, 257-260
- [37] Bunke H., Allermann G.: *On a relation between graph edit distance and maximum common subgraph*, Pattern Recognition Letters, Vol. 18, 1997, 689-694
- [38] Bunke H.: *Recent developments in graph matching*, Proc. of the 15-th Int. Conf. on Pattern Recognition, Vol. 2, Barcelona/Spain, 2000, 117 - 124
- [39] Bunke H.: *Graph matching: Theoretical foundations, algorithms, and applications*, Proc. Vision Interface 2000, Montreal/Canada, 2000, 82 - 88
- [40] Bush V.: *As we may think*, The Atlantic Monthly, Vol. 176 , 1945, 101-108
- [41] Buttler D.: *A short survey of document structure similarity algorithms*, Proceedings of the International Conference on Internet Computing, Las Vegas/USA, 2004, 3-9
- [42] Causton H. C., Brazma A., Quackenbush J.: *Microarray Gene Expression Data Analysis*, Blackwell Publishers, 2003
- [43] Carrière J., Kazman R.: *WebQuery: Searching and visualizing through connectivity*, Proc. of the 6-th WWW Conf., 1997

- [44] Chakrabarti S., Dom B., Indyk P.: *Enhanced hypertext categorization using hyperlinks*, Proceedings of ACM SIGMOD International Conference on Management of Data, 1998, 30-318
- [45] Chakrabarti S.: *Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction*, Proc. of the 10-th International World Wide Web Conference, Hong Kong, 2001, 211-220
- [46] Chakrabarti S.: *Mining the Web. Discovering Knowledge from Hypertext Data*, Morgan and Kaufmann Publishers, 2003
- [47] Caley A.: *On the analytical forms called trees, with application to the theory of chemical combinatorics*, Report of the British Association for the Advancement of Science, 1875, 257-305
- [48] Charney D.: *Comprehending non-linear text: The role of discourse cues and reading strategies*, Proc. Hypertext'87, Chapel Hill/USA, 1987, 109-120
- [49] Chen W. K.: *Applications and realizability of degree sequences of directed graphs without parallel edges and self-loops*, The Matrix and Tensor Quarterly, Vol. 24, 123-130, 1974, Vol. 25, 1974, 1-7
- [50] Chomsky N.: *Aspekte der Syntax-Theorie*, Suhrkamp Verlag, 1972
- [51] Christen H. R., Meyer G.: *Grundlagen der allgemeinen und anorganischen Chemie*, Diesterweg Verlag, 1997
- [52] Chung Fan R. K.: *Spectral Graph Theory*, Cbms Regional Conference Series in Mathematics. Amer. Mathematical Society, Vol. 92, 1997
- [53] Cooley R., Srivastava J., Mobasher B.: *Web mining: Information and pattern discovery on the world wide web*, Proceedings of the 9-th IEEE International Conference on Tools with Artificial Intelligence, 1997
- [54] Conclin J.: *Hypertext: An introduction and survey*, IEEE Computer, Vol. 20 (9), 1987, 17-41
- [55] Coulston C., Vitolo T. M.: *A hypertext metric based on huffman coding*, Proc. ACM Conf. of Hypertext, 2001, 243-244
- [56] Cristianini N., Shawe-Taylor J.: *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK, 2000
- [57] Cruz I. F., Borisov S., Marks M. A., Webb T. R.: *Measuring structural similarity among web documents: Preliminary results*, Lecture Notes in Computer Science, Vol. 1375, 1998

- [58] Cvetkovic D. M., Doob M., Sachs H.: *Spectra of Graphs. Theory and Application*, Academic Press, 3-rd Edition, 1997
- [59] De Bra P., Houben G. J.: *Hypertext metrics revisited: Navigational metrics for static and adaptive link structures*, <http://citeseer.ist.psu.edu/139855.html>
- [60] De Bra P.: *Using hypertext metrics to measure research output levels*, Scientometrics, Vol. 47 (2), Kluwer Academic Publishers, 2000, 227-236
- [61] Dehmer M., Gleim R., Mehler A.: *Aspekte der Kategorisierung von Webseiten*, GI-Edition - Lecture Notes in Informatics (LNI) - Proceedings, Jahrestagung der Gesellschaft für Informatik, Informatik 2004, Ulm/Germany, 2004, 39-43
- [62] Dehmer M., Gleim R., Mehler A. : *A new method of measuring similarity for a special class of directed graphs*, Tatra Mountains Mathematical Publications, Slovakia, eingereicht zur Publikation, August 2004
- [63] Dehmer M.: *Data Mining-Konzepte und graphentheoretische Methoden zur Analyse hypertextueller Daten*, erscheint in: LDV Forum, Zeitschrift für Computerlinguistik und Sprachtechnologie, 2005
- [64] Delisle N. M., Schwartz M.: *Neptune - A hypertext system for software development environment*, Database Engineering, Vol. 10 (1), 1987, 54-59
- [65] Deo N., Gupta P.: *World wide web: A graph-theoretic perspective*, Computer Science Technical Report; University of Central Florida, 2001
- [66] Diestel R.: *Graphentheorie*, Springer Verlag, Berlin-Heidelberg, 2000
- [67] d'Inverno M., Priestley M., Luck M.: *A formal framework for hypertext systems*, IEEE Proceedings on Software Engineering, Vol. 144 (3), 1997, 175-184
- [68] Duda R. O., Hart P. E., Stork D. G.: *Pattern Classification*, Wiley - Interscience, 2000
- [69] Ehud R., Botafogo R. A., Shneiderman B.: *Navigating in hyperspace: Designing a structure-based toolbox*, Commun. ACM 37 (2), 1994, 87-96
- [70] Ehrig H.: *Introduction to the algebraic theory of graph grammars*, Lecture Notes in Computer Science, Proceedings of the International Workshop on Graph-Grammars and their Application to Computer Science and Biology, 1978, 1-69

- [71] Ehrig H.: *Introduction to graph grammars with applications to semantic networks*, Computers and Mathematics with Applications, Vol. 23, 1992, 557-572
- [72] Eiron D., McCurly K. S.: *Untangling compound documents on the web*, Proceedings of the 14-th ACM Conference on Hypertext and Hypermedia, Nottingham/UK, 2003, 85-94
- [73] Emmert-Streib F., Dehmer M., Kilian J.: *Classification of large graphs by a local tree decomposition*, erscheint in: Proceedings of DMIN'05, International Conference on Data Mining, In Verbindung mit: World Congress in Applied Computing 2005, Las Vegas/USA, Juni 20-23, 2005
- [74] Englebart D.: *Augmenting the human intellect: A conceptual framework*, Stanford Research Institute, Menlo Park/USA, 1962
- [75] Erdős P.: *Graph theory and probability II*, Canad. J. Math., Vol 13, 1961, 346-352
- [76] Erdős P.: *Some applications of probability to graph theory and combinatorial problems*, Theory Graphs Appl., Proc. Symp. Smolenice/CSSR 1963, 1964, 133-136
- [77] Euler L.: *Solutio problematis ad geometriam situs pertinentis*, Commentarii Academiae Petropolitanae, Vol. 8, 1736, 128-140
- [78] Everitt B. S., Landau S., Leese M.: *Cluster Analysis*, Arnold Publishers; 4-th Edition, 2001
- [79] Fasulo D.: *An analysis of recent work on clustering algorithms*, Technical Report 01-03-02, University of Washington, Seattle/USA, 1999
- [80] Fayyad U. M., Piatetsky-Shapiro G., Smyth P.: *From data mining to knowledge discovery: An overview*, In: Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press, Menlo Park, Editors: Fayyad U. M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R., 1996, 1-34
- [81] Ferber R.: *Information Retrieval*, dpunkt.verlag, Heidelberg, 2003
- [82] Fichtenholz G. M.: *Differential- und Integralrechnung*, VEB Deutscher Verlag der Wissenschaften, 1964
- [83] Fischer G.: *Lineare Algebra*, Vieweg Verlag, 2003
- [84] Flesca S., Manco G., Masciari E., Pontieri L., Pugliese A.: *Detecting structural similarities between XML documents*, 5-th International Workshop on the Web and Databases, 2002

- [85] Fronk A.: *Algebraische Semantik einer objektorientierten Sprache zur Spezifikation von Hyperdokumenten*, Dissertation, Universität Dortmund, Fachbereich Informatik, Lehrstuhl Software-Technologie, Shaker Verlag, 2002
- [86] Fronk A.: *Towards the algebraic analysis of hyperlink structures*, International Journal on Software Engineering and Knowledge Engineering, Vol. 13 (6), 2003, 655-684
- [87] Fürnkranz J.: *Hyperlink Ensembles: A Case Study in Hypertext Classification*, Technical Report OEFAI-TR-2001-30, Austrian Research Institute for Artificial Intelligence, Wien/Austria
- [88] Furner J., Ellis D., Willet P.: *The representation and comparison of hypertext structures using graphs*, In: Information Retrieval and Hypertext, Kluwer, Boston, 1997, 75-96
- [89] Foulds L. R.: *Graph Theory Applications*, Springer Verlag, New York, 1992
- [90] Gärtner T., Flach P. A., Wrobel S.: *On graph kernels: Hardness results and efficient alternatives*, 17-th Annual Conference on Computational Learning Theory and 7-th Kernel Workshop (COLT-2003), Springer, 2003
- [91] Gallo G., Longo G., Pallottino S., Nguyen S.: *Directed hypergraphs and applications*, Discrete Appl. Math., Vol. 42 (2-3), 1993, 177-201
- [92] Gernert D.: *Measuring the similarity of complex structures by means of graph grammars*, Bulletin of the EATCS, Vol. 7, 1979, 3-9
- [93] Gernert D.: *Graph grammars which generate graphs with specified properties*, Bulletin of the EATCS, Vol. 13, 1981, 13-20
- [94] Gleim R.: *Integrierte Repräsentation, Kategorisierung und Strukturanalyse Web-basierter Hypertexte*, Diplomarbeit der Technischen Universität Darmstadt, Fachbereich Informatik, September 2004
- [95] Gleim R.: *HyGraph – Ein Framework zur Extraktion, Repräsentation und Analyse webbasierter Hypertextstrukturen*, In: Sprachtechnologie, mobile Kommunikation und linguistische Ressourcen. Beiträge zur GLDV-Tagung 2005 in Bonn, Editors: Fisseni B., Schmitz H. S., Schröder B., Wagner P., Peter Lang Verlag, Frankfurt am Main
- [96] Glöggl M.: *Suchmaschinen im Internet*, Springer, 2003
- [97] Gregson A. M. R.: *Psychometrics of Similarity*, Academic Press, New York, 1975

- [98] Godsil C., Royle G.: *Algebraic Graph Theory*, Graduate Texts in Mathematics, Springer-Verlag, 2001
- [99] Gusfield D.: *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, 1997
- [100] Halasz F. G.: *Reflections on NoteCards: Seven issues for the next generation of hypermedia systems*, Proc. Hypertext'87, Chapel Hill/USA, 1987, 345-366
- [101] Hakimi S. L.: *On the realizability of a set of integers as degrees of a graph*, J. SIAM Appl. Math., Vol. 10, 1962, 496-506
- [102] Hakimi S. L.: *On the degrees of the vertices of a directed graph*, J. Franklin Inst., Vol. 279, 1965, 290-308
- [103] Halin R.: *Graphentheorie*, Akademie Verlag Berlin, 1989
- [104] Han J., Kamber M.: *Data Mining: Concepts and Techniques*, Morgan and Kaufmann Publishers, 2001
- [105] Harary F.: *Status and contrastatus*, Sociometry, Vol. 22, 1959, 23-43
- [106] Harary F.: *Structural models. An introduction to the theory of directed graphs*, Wiley, New York, 1965
- [107] Harary F., Palmer E. M.: *Graphical Enumeration*, Academic Press, New York, 1973
- [108] Harary F.: *Graphentheorie*, Oldenbourg Verlag, München, 1974
- [109] Hastie R., Tibshirani R., Friedman J. H.: *The Elements of Statistical Learning*, Springer Verlag, 2001
- [110] Hearst M. A., Schölkopf B., Dumais S., Osuna E., Platt J.: *Trends and controversies - Support Vector Machines*, IEEE Intelligent Systems, Vol. 13 (4), 1998, 18-28
- [111] Heuser H.: *Lehrbuch der Analysis. Teil 1*, Teubner Verlag Stuttgart, 1991
- [112] Höchstmann M., Töller T., Giegerich R., Kurtz S.: *Local similarity in RNA secondary structures*, Proceedings of the IEEE Computational Systems Bioinformatics Conference (CSB'03), 2003, 159-168
- [113] Hofmann M.: *Benutzerunterstützung in Hypertextsystemen durch private Kontexte*, Dissertation, Technische Universität Carolo-Wilhemina Braunschweig, 1991

- [114] Horney M.: *A measure of hypertext linearity*, Journal of Educational Multimedia and Hypermedia, Vol. 2 (1), 1993, 67-82
- [115] Horváth T., Gärtner T., Wrobel S.: *Cyclic pattern kernels for predictive graph mining*, Proceedings of the 2004 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, 158-167
- [116] Horváth T.: *Cyclic pattern kernels revisited*, In: Advances in Knowledge Discovery and Data Mining, Proceedings of the 9-th Pacific-Asia Conference, PAKDD 2005, Hanoi/Vietnam, 2005, Editors: Ho T. B, Cheung D., Liu H., 2005, 791-801
- [117] Huberman B., Adamic L.: *Growth dynamics of the world-wide web*, Nature, Sept. 1999, 401-131
- [118] Huffman D. A.: *A method for the construction of minimum-redundancy codes*, Proceedings of the I.R.E., Vol. 40 (9), 1952, 1098-1101
- [119] Hsu C. W., Chang C. C., Lin C. J.: *A practical guide to SVM classification*, Technical report, Department of Computer Science and Information Technology, National Taiwan University, 2003.
- [120] Ihringer T.: *Diskrete Mathematik*, Teubner Verlag, 1994
- [121] Inokuchi A., Washio T., Motada H.: *Complete Mining of frequent patterns from graphs: Mining graph data*, Machine Learning, Vol. 50 (3), 2003, 321-354
- [122] Jain A. K., Dubes R. C.: *Algorithms for Clustering Data*, Prentice Hall, 1988
- [123] Jänich K.: *Topologie*, Springer Verlag, 1999
- [124] Jiang T., Wang L., Zhang K.: *Alignment of trees - An alternative to tree edit*, Theoretical Computer Science, Elsevier, Vol. 143, 1995, 137-148
- [125] Joachims T.: *Learning to classify text using support vector machines*, Kluwer, Boston, 2002
- [126] Jolion J. M.: *Graph matching: What we are really talking about?*, Third IAPR Workshop on Graph-based Representations in Pattern Recognition, Ischia/Italy, 2001
- [127] Joshi S., Agrawal N., Krishnapuram R., Negi S.: *Bag of paths model for measuring structural similarity in web documents*, Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2003, 577-582

- [128] Kähler W. M.: *Statistische Datenanalyse. Verfahren verstehen und mit SPSS gekonnt einsetzen*, Vieweg Verlag, 2002
- [129] Kaden F.: *Graphmetriken und Distanzgraphen*, ZKI-Informationen, Akad. Wiss. DDR, 1982, 1-63
- [130] Kaden F.: *Halbgeordnete Graphmengen und Graphmetriken*, In: *Graphs, Hypergraphs, and Applications*, Editor: Sachs H., Teubner-Texte zur Mathematik, Vol. 73, Leipzig, 1983, 92-95
- [131] Kaden F.: *Graphmetriken und Isometrie Probleme zugehöriger Distanzgraphen*, ZKI-Informationen, Akad. Wiss. DDR, 1986, 1-100
- [132] Kajitanai Y., Sakurai H.: *On distance of graphs defined by use of orthogonality between circuits and cutsets*, Rep. Inst. Electron. Comm. Engineers, Japan, 1973
- [133] Kajitanai Y., Ueda M.: *On the metric space of labeled graphs*, Rep. Inst. Electron. Comm. Engineers, Japan, 1975
- [134] Kirchhoff G.: *Über die Auflösung von Gleichungen, auf welche man bei der Untersuchung der linearen Verteilungen galvanischer Ströme geführt wird*, Annalen der Physik und Chemie, Vol. 72, 1847, 497-508
- [135] Kleinberg J. M.: *Authoritative sources in a hyperlinked environment*, Proceedings of the 9-th Annual ACM-SIAM Symposium on Discrete Algorithms, 1998, 668-677
- [136] Kobayashi M., Takeda K.: *Information retrieval on the web*, ACM Comput. Surv., Vol. 32 (2), 2000, 144-173
- [137] König D.: *Theorie der endlichen und unendlichen Graphen*, Chelsea Publishing, New York, 1935
- [138] Kommers P. A. M.: *Hypertext and acquisition of knowledge*, Ph.D-Thesis, Enschede/Netherlands, 1990
- [139] Kosala R., Blockeel H.: *Web mining research: A survey*, SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery & Data Mining, ACM, Vol. 2, 2000
- [140] Kuhlen R.: *Hypertext: Ein nicht-lineares Medium zwischen Buch und Wissensbank*, Springer Verlag, 1991
- [141] Kumar R., Raghavan P., Rajagopalan S., Sivakumar D., Tomkins A., Upfal E.: *The web as a graph*, Proceedings of the 19-th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, Dallas/USA, 2000



- [142] Kumar R., Raghavan P., Rajagopalan S., Sivakumar D.: *Stochastic graph models for the web graph*, Proceedings of the 41-th Annual Symposium on Foundations of Computer Science, Redondo Beach/USA, 2000, 12-14
- [143] Kuratowski K.: *Sur le problème des courbes gauches en topologie*, Fund. Math. Vol. 15, 1930 271-283
- [144] Lange D. B.: *A formal model of hypertext*, NIST Hypertext Standardization Workshop, 1990, 145-166
- [145] Lesk A. M.: *Bioinformatik. Eine Einführung*, Spektrum Akademischer Verlag, 2003
- [146] Levenstein V. I.: *Binary codes capable of correcting deletions, insertions and reversals*, Soviet Phys. Dokl., Vol. 10, 1966, 707-710
- [147] Li M., Chen X., Li X., Ma B., Vitany P. M. D.: *The similarity metric*, Proceedings of the 14-th Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, 863-872,
- [148] Liebetrau A. M.: *Measures of association*, Sage University Paper Series on Quantitative Applications in the Social Sciences - Sage Publications, Beverley Hills, 1983
- [149] Lipman D., Pearson W.: *Improved tools for biological sequence comparison*, Fund. Math., Vol. 15, 1930, 271-283
- [150] Lobin H.: *Text im digitalen Medium. Linguistische Aspekte von Textdesign, Texttechnologie und Hypertext Engineering*, Westdeutscher Verlag, Wiesbaden/Germany, 1999
- [151] McEneaney J. E.: *Visualizing and assessing navigation in hypertext*, Proc. ACM Conf. of Hypertext, 1999, 61-70
- [152] McEneaney J. E.: *Navigational correlates of comprehension in hypertext*, Proc. ACM Conf. of Hypertext, 2000, 251-255
- [153] Mehler A.: *Textbedeutung. Zur prozeduralen Analyse und Repräsentation struktureller Ähnlichkeiten von Texten*, Peter Lang, Europäischer Verlag der Wissenschaften, 2001
- [154] Mehler A.: *Hierarchical orderings of textual units*, Proc. of COLING'02, 2002
- [155] Mehler A.: *Textmining*, In: Texttechnologie. Perspektiven und Anwendungen, Editor: Lobin H., Stauffenburg Verlag, 2004

- [156] Mehler A., Dehmer M., Gleim R.: *Towards logical hypertext structure. A graph-theoretic perspective*, Proc. of I2CS'04, Guadalajara/Mexico, Berlin-New York: Springer, 2004
- [157] Mehler A., Dehmer M., Gleim R.: *Automatische Klassifikation von Webgenre-Graphen*, GLDV Frühjahrstagung, Universität Bonn, 2005
- [158] Mehler A., Gleim R., Dehmer M.: *Towards structure-sensitive hypertext categorization*, erscheint in: Proceedings of the 29-th Annual Conference of the German Classification Society, 2005
- [159] Mühlhäuser M.: *Hypermedia-Konzepte zur Verarbeitung multimedialer Information*, Informatik-Spektrum Vol. 14 (5), Special Issue on Multimedia, 1991, 281-290
- [160] Mühlhäuser M.: *eLearning After Four Decades: What About Sustainability?*, Proceedings of ED-MEDIA 2004, 3694-3700
- [161] Nagl M.: *Graph-Grammatiken: Theorie, Anwendungen, Implementierung*, Vieweg-Verlag, 1979
- [162] Needleman S. B., Wunsch C. D.: *A general method applicable to the search for similarities in the amino acid sequences of two proteins*, Journal of Molecular Biology, Vol. 48, 1970, 443-453
- [163] Nehmhauser G. L.: *Einführung in die Prozesse der dynamischen Programmierung*, Oldenbourg Verlag, 1969
- [164] Nelson T.: *Computer Lib/Dream Machines*, Hugo Press, Chicago, 1974
- [165] Nelson T.: *All for one and one for all*, Vorartikel der Proc. Hypertext'87, Chapel Hill/USA, 1987
- [166] Nielsen J.: *Hypertext and Hypermedia*, Academic Press Professional, Cambridge-Massachusetts, 1993
- [167] Nielsen J.: *Multimedia, Hypertext und Internet. Grundlagen und Praxis des elektronischen Publizierens*, Vieweg Verlag, Wiesbaden/Germany, 1996
- [168] Noller S., Naumann J., Richter T.: LOGPAT - *ein webbasiertes Tool zur Analyse von Navigationsverläufen in Hypertexten*, In: Perspectives on internet research: Concepts and methods, Editors: Jonas K. J., P. Breuer P., Schauenburg B., Boos M., <http://www.psych.uni-goettingen.de/congress/gor-2001>, 2001
- [169] Novak L., Gibbons A., van Rijsbergen C. J.: *Hybrid Graph Theory and Network Analysis*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 1999

- [170] Oommen B. J., Zhang K., Lee W.: *Numerical similarity and dissimilarity measures between two trees*, IEEE Transactions on Computers, Vol. 45 (12), 1996, 1426-1435
- [171] Oren T.: *The architecture of static hypertext*, Proc. Hypertext'87, Chapel Hill/USA, 1987, 291-306
- [172] Palmer C., Gibbons P., Faloutsos C.: *ANF: A fast and scalable tool for data mining in massive graphs*, Proc. of the 8th ACM SIGKDD Internal Conference on Knowledge Discovery and Data Mining, 2002
- [173] Park S.: *Structural Properties of Hypertext*, Proc. ACM Conf. of Hypertext, 180-187, 1998
- [174] Parunak V. D. H.: *Don't link me in: Set based hypermedia for taxonomic reasoning*, Proc. ACM Conf. of Hypertext, Hypertext'91, 1991, 233-242
- [175] Petersen J.: *Die Theorie der regulären Graphs*, Acta Mathematica Vol. 15, 1891, 193-220
- [176] Raghavan V., Bollmann P., Jung G.: *Critical investigation of recall and precision as measures of retrieval system performance*, ACM Transactions on Information Systems, Vol. 7 (3), 1989, 205-229
- [177] Raghavan P.: *Graph structure of the web: A survey*, LATIN 2000: Theoretical Informatics. 4 th Latin American Symposium, 2000, 123-125
- [178] Rahm E.: *Web usage mining*, Datenbank-Spektrum, Vol. 2 (2), 2002, 75-76
- [179] Richter T., Naumann J., Noller S.: *LOGPAT: A semi-automatic way to analyze hypertext navigation behavior*, Swiss Journal of Psychology, Vol. 62, 2003, 113-120.
- [180] Rieger B.: *Unschärfe Semantik*, Peter Lang Verlag, 1989
- [181] Robertson N., Seymour P. D.: *Graph minors II. Algorithmic aspects of tree-width*, Journal of Algorithms, Vol. 7, 1986, 309-322
- [182] Robles-Kelly A., Hancock R.: *Edit distance from graph spectra*, Proceedings of the IEEE International Conference on Computer Vision, 2003, 234-241
- [183] Rückert U., Kramer S.: *Frequent free tree discovery in graph data*, Proceedings of the 2004 ACM Symposium on Applied Computing, 2004, 564-570
- [184] Rühls F.: *Funktionentheorie*, VEB Deutscher Verlag der Wissenschaften, Berlin, 1976

- [185] Ruskey F., Eades P., Cohen B., Sedgwick A.: *Alley CATs in search of good homes*, 25-th S.E. Conference on Combinatorics, Graph Theory, and Computing, Congressus Numerantium, Vol. 102, 1994, 97-110
- [186] Sachs H., Finck H. J., Hutschenreuther H., Kaiser H., Lang R., Schäuble M., Voß H. J., Walther H.: *Einführung der Theorie der endlichen Graphen*, Carl Hanser Verlag, München, 1971
- [187] Sachs H.: *Einführung der Theorie der endlichen Graphen II*, Mathematisch Naturwissenschaftliche Bibliothek 44, 1972
- [188] Sanfeliu A., Fu K. S., Prewitt J. M. S.: *An application of a distance measure between graphs to the analysis of muscle tissue patterns*, Workshop on Structural Pattern and Dyntactic Pattern Recognition, Saratoga-Springs/USA, 1981, 86-89
- [189] Sanfeliu A., Fu K. S.: *A distance measure between attributed relational graphs for pattern recognition*, IEEE Transactions on Systems, Man, and Cybernetics, 1983, 353-362
- [190] Sankoff D., Kruskal J. B.: *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley, 1983
- [191] Sankoff D., Kruskal J. B., Mainville S., Cedergren R. J.: *Fast algorithms to determine RNA secondary structures containing multiple loops*, In: Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison, Editors: Sankoff D. und Kruskal J. B., Addison-Wesley, 1983, 93-120
- [192] Schäuble P.: *Multimedia Information Retrieval. Content-Based Informations Retrieval from Large Text and Audio Databases*, Kluwer Academic Publishers, 1997
- [193] Schlobinski P., Tewes M.: *Graphentheoretische Analyse von Hypertexten*, NETWORX - Online-Publikationen zum Thema Sprache und Kommunikation im Internet, <http://www.websprache.uni-hannover.de/networx/docs/networx-8.pdf>
- [194] Schmidt G., Ströhlein T.: *Relationen und Graphen*, Wiley VCH, 2002
- [195] Schnupp P.: *Hypertext*, Oldenbourg Verlag, 1992
- [196] Schölkopf B., Müller K. R., Smola A. J.: *Lernen mit Kernen. Support-Vektor-Methoden zur Analyse hochdimensionaler Daten*, Informatik Forsch. Entw., Vol. 14, 1999, 154-163

- [197] Schönefeld T.: *Bedeutungskonstitution in Hypertext*, NETWORKX - Online-Publikationen zum Thema Sprache und Kommunikation im Internet, <http://www.websprache.net/networkx/docs/networkx-19.pdf>
- [198] Schöning U.: *Graph isomorphism is in the low hierarchy*, Symposium on Theoretical Aspects of Computer Science, 1987, 114-124.
- [199] Schöning U.: *Algorithmen - kurz gefasst*, Spektrum Akademischer Verlag, 1997
- [200] Schöning U.: *Theoretische Informatik - kurz gefasst*, Spektrum Akademischer Verlag, 2001
- [201] Schulmeister R.: *Grundlagen hypermedialer Lernsysteme*, Oldenbourg Verlag, 2002
- [202] Schubert H.: *Topologie*, Teubner Verlag, Stuttgart, 1971
- [203] Schulz H. J.: *Visuelles Data Mining komplexer Strukturen*, Diplomarbeit der Universität Rostock, Institut für Informatik, 2004
- [204] Scott F.: *Social Network Analysis*, Sage Publications, 2001
- [205] Selkow S. M.: *The tree-to-tree editing problem*, Information Processing Letters, Vol. 6 (6), 1977, 184-186
- [206] Sernetz M.: *Fraktale biologische Strukturen: Chaos und Ordnung im Organismus*, Berichte der Justus Liebig-Gesellschaft zu Gießen e.V., Vol. 5, 2001, 143-158
- [207] Shapiro L.: *Organization of relational models for scene analysis*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 4, 1982, 595-603
- [208] Shapiro L.: *Organization of relational models*, Proc. Intern. Conf. on Pattern Recognition, München, 1982, 360-365
- [209] Shapiro B. A., Zhang K.: *Comparing multiple RNA secondary structures using tree comparisons*, Comp. Appl. Biosci., Vol. 6 (4), 1990, 309-318
- [210] Smith J. B., Weiss S. F., Ferguson G. J.: *A hypertext environment and its cognitive basis*, Proc. Hypertext'87, Chapel Hill/USA, 1987, 195-214
- [211] Shneiderman B., Kearsley G.: *Hypertext Hands On!: An introduction to a new way of organizing and accessing information*, Addison Wesley, 1989

- [212] Skvortsova M. I., Baskin I. I., Stankevich I. V., Palyulin V. A., Zefirov N. S.: *Molecular similarity in structure-property relationship studies. Analytical description of the complete set of graph similarity measures*, International symposium CACR-96. Book of Abstracts. December 17-18, Moskau/Russia, 1996
- [213] Sobik F.: *Graphmetriken und Klassifikation strukturierter Objekte*, ZKI-Informationen, Akad. Wiss. DDR, Vol. 2 (82), 1982, 63-122
- [214] Sobik F.: *Modellierung von Vergleichsprozessen auf der Grundlage von Ähnlichkeitsmaßen für Graphen*, ZKI-Informationen, Akad. Wiss. DDR, Vol. 4, 1986, 104-144
- [215] Späth H.: *Cluster - Analyse - Algorithmen*, Oldenbourg Verlag, 1977
- [216] Spertus E.: *ParaSite: Mining structural information on the web*, Proc. of the 6-th WWW Conf., 1997
- [217] Spiliopoulou M.: *Web usage mining for web site evaluation*, Comm. ACM, Vol. 43 (8), 2000, 127-134
- [218] Steinhausen D., Langer K.: *Clusteranalyse. Einführung in Methoden und Verfahren der automatischen Klassifikation*, Walter de Gruyter Verlag, 1997
- [219] Steinmetz R.: *Multimedia - Technologie*, Springer Verlag, 2000
- [220] Steinmetz R., Nahrstedt K.: *Multimedia Systems*, Springer Verlag, 2004
- [221] Storrer A.: *Kohärenz in Text und Hypertext*, In: Text im digitalen Medium. Linguistische Aspekte von Textdesign, Texttechnologie und Hypertext Engineering, Editor: Lobin H., Westdeutscher Verlag, Wiesbaden/Germany, 1999, 33-65
- [222] Storrer A.: *Text und Hypertext*, In: Texttechnologie. Perspektiven und Anwendungen, Editor: Lobin H., Stauffenburg Verlag, 2004
- [223] Stotts P. D., Furuta R.: *Petri-Net based hypertext: Document structures with browsing semantics*, ACM Transactions of Information Systems, Vol. 7 (1), 1989, 3-29
- [224] Struyf A., Hubert M., Rousseeuw P. J.: *Clustering in an object-oriented environment*, Journal of Statistical Software, Vol. 1 (4), 1996
- [225] Sylvester J. J.: *On an application of the new atomic theory to the graphical representation of the invariants and covariants of binary quantics*, American Journal of Mathematics, Vol. 1, 64-125, 1878

- [226] Tai K.: *The tree-to-tree correction problem*, Journal of the ACM, Vol. 26, 1979, 422-433
- [227] Tanaka E.: *A metric on graphs and its application*, Technical Report, IEE Japan, 1977
- [228] Tiedge D.: *Rezeption von Text und Hypertext - ein Vergleich*, Linguistik-Server Essen, [http://www.linse.uni-essen.de/esel/pdf/rezeption\\_hypertexte.pdf](http://www.linse.uni-essen.de/esel/pdf/rezeption_hypertexte.pdf)
- [229] Tittmann P.: *Graphentheorie*, Springer Verlag, Berlin-Heidelberg-New York, 1989
- [230] Tochtermann K., Dittrich G.: *The dortmund family of hypermedia models - concepts and their application*, Journal of Universal Computer Science, Vol. 2 (1), 1996, 34-56
- [231] Tompa F.: *A data model for flexible hypertext database systems*, ACM Top. Inf. Syst. Vol. 7, 1989, 85-100
- [232] Turau V.: *Algorithmische Graphentheorie*, Oldenbourg Verlag, 1996
- [233] Tversky A.: *Features of similarity*, Psychological Review, Vol. 84 (4), 327-352, 1977
- [234] Ullman J. R.: *An algorithm for subgraph isomorphism*, J. ACM, Vol. 23 (1), 1976, 31-42
- [235] Unz D.: *Lernen mit Hypertext. Informationsuche und Navigation*, Waxmann Verlag, 2000
- [236] Vapnik V.: *The Nature of Statistical Learning Theory*, Springer Verlag, 1995
- [237] Veblen O.: *Analysis Situs*, Amer. Math. Soc. Colloq. Publ., Vol. 5, Cambridge, 1922. 2-te Auflage, New York, 1931
- [238] Vogt J.: *Hypertext. Die neue Form der Schriftlichkeit?*, Magisterarbeit, Universität Stuttgart, Institut für Literaturwissenschaft, 2000
- [239] Volkmann L.: *Graphen und Digraphen. Eine Einführung in die Graphentheorie*, Springer Verlag, Wien, 1991
- [240] Wang L., Zhao J.: *Parametric alignments of ordered trees*, Bioinformatics, Vol. 19 (17), 2003, 2237-2245
- [241] Washio T., Motoda H.: *State of the art of graph-based data mining*, ACM SIGKDD Explorations Newsletter, Vol. 5 (1), 2003, 59-68

- [242] Watts D.: *Small Worlds: The Dynamics of Networks between Order and Randomness*, Princeton University Press, Princeton, 1999
- [243] Watts D., Strogatz S.: *Collective dynamics of small-world networks*, Nature, Vol. 393, 1998, 440-442
- [244] Weiss Y.: *Segmentation using eigenvectors: A unifying view*, Proc. 7-th Int. Conf. Computer Vision, 1999, 975-982.
- [245] Wiener H.: *Structural determination of paraffin boiling points*, Journal of the American Chemical Society, Vol. 69, 1947, 17-20
- [246] Wilhelm R., Heckmann R.: *Grundlagen der Dokumentenverarbeitung*, Addison Wesley, 1996
- [247] Winne P. H., Gupta L., Nesbit J. C.: *Exploring individual differences in studying strategies using graph theoretic statistics*, The Alberta Journal of Educational Research, Vol. 40, 177-193, 1994
- [248] Winter A.: *Exchanching graphs with GXL*, <http://www.gupro.de/GXL>
- [249] Witten I, Eibe F.: *Data Mining*, Hanser Fachbuchverlag, 2001
- [250] Wrobel S., Morik K., Joachims T.: *Maschinelles Lernen und Data Mining*, In: Handbuch der Künstlichen Intelligenz, Oldenbourg-Verlag, Editors: Görz G., Rollinger C., Schneeberger J., 2000, 517-597
- [251] Yang Y., Pedersen J. O.: *A comparative study on feature selection in text categorization*, Proceedings of the 14-th International Conference on Machine Learning, 1997, 412-420
- [252] Yang Y., Slattey S., Ghani R.: *A study of approaches to hypertext categorization*, Journal of Intelligent Information Systems, Vol. 18 (2-3), 2002, 219-241
- [253] Xifeng Yan J. H.: *gSpan: Graph-based substructure pattern mining*, Proceedings of the 2002 IEEE International Conference on Data Mining, 2002, 721-724
- [254] Zelinka B.: *On a certain distance between isomorphism classes of graphs*, Časopis pro pest. Matematiky, Vol. 100, 1975, 371-373
- [255] Zhang K., Shasha D.: *Simple fast algorithms for the editing distance between trees and related problems*, SIAM Journal of Computing, Vol. 18 (6), 1989, 1245-1262
- [256] Zhang K., Statman R., Shasha D.: *On the editing distance between unordered labeled trees*, Information Processing Letters, Vol. 42 (3), 1992, 133-139



[257] *HTMLParser* - *Sourceforge*, <http://htmlparser.sourceforge.net>

[258] *SELFHTML*, <http://de.selfhtml.org>



# Lebenslauf

## Persönliche Daten

Name: Matthias Dehmer  
Geburtsdatum: 16.04.1968  
Geburtsort: Wiesbaden

## Schulausbildung

1974 - 1978 Grundschole in Taunusstein-Wehen  
1978 - 1985 Gesamtschole in Taunusstein-Hahn  
1985 Mittlere Reife  
1991 - 1993 HBFS I in Mainz  
1993 Phys.-techn.-Assistent incl. Fachabitur

## Ausbildung und Berufstätigkeit I

1985 - 1986 BGJ für Metall- und Agrartechnik/Wiesbaden  
1986 - 1989 Metallbauerausbildung bei der Fa.Lerner/Wiesbaden  
1989 - 1991 Schweißer bei der Fa. Ülzener, Fa. Schiele/Frankfurt  
1993 - 1994 Phys.-techn.-Assistent bei der Fa. Linde/Unterschleissheim

## Hochschulausbildung

WS 1994 - SS 1998 Mathematikstudium an der Universität-GH Siegen  
SS 1998 Diplom-Mathematiker

## Berufstätigkeit II

01.1999 - 03.2000 Dipl.-Math. bei der AL-Versicherung/Oberursel  
04.2000 - 10.2001 Schulungsleiter bei der Sybase GmbH/Frankfurt  
11.2001 - 04.2002 Kompetenzmanager bei der Siemens AG/Frankfurt  
05.2002 - 07.2005 Wiss. Mitarbeiter und Doktorand  
im Fachbereich Informatik der TU-Darmstadt  
(Prof. Dr. Max Mühlhäuser)



# Erklärung

Hiermit erkläre ich, die vorliegende Arbeit zur Erlangung des akademischen Grades Dr. rer. nat. mit dem Titel „Strukturelle Analyse Web-basierter Dokumente“ selbstständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keine Promotionsversuche unternommen.

Darmstadt, den 25.07.2005

MATTHIAS DEHMER